Kauno technologijos universitetas

Informatikos fakultetas

Autonominių transporto priemonių interneto srauto šifravimo metodas

Baigiamasis magistro projektas

Mantas Obolevičius

Projekto autorius

Prof. Algimantas Venčkauskas

Vadovas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Autonominių transporto priemonių interneto srauto šifravimo metodas

Baigiamasis magistro projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Mantas Obolevičius

Projekto autorius

prof. Algimantas Venčkauskas

Vadovas

doc. Tomas Adomkus

Recenzentas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Mantas Obolevičius

Autonominių transporto priemonių interneto srauto šifravimo metodas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Mantas Obolevičius

Patvirtinta elektroniniu būdu

Obolevičius, Mantas. Autonominių transporto priemonių interneto srauto šifravimo metodas. Magistro krypties studijų baigiamasis projektas / vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos inžinerija (Informatikos mokslai).

Reikšminiai žodžiai: autonominės transporto priemonės, šifravimas, virtualus privatus tinklas, WireGuard, TPM, TLS, saugus ryšys, šifruotas duomenų kanalas, V2X komunikacija, tinklo saugumas.

Kaunas, 2025. 65 p.

Santrauka

Pagrindinis darbo tikslas yra sukurti ir įvertinti saugų interneto srauto šifravimo metodą, skirtą autonominių transporto priemonių ir kelio infrastruktūros objektų tarpusavio komunikacijai. Darbe analizuojamos šiuolaikinės tinklo šifravimo technologijos, virtualaus privataus tinklo protokolai bei jų tinkamumas realaus laiko transporto sistemoms. Ypatingas dėmesys skiriamas efektyvumui, saugumui ir diegimo paprastumui.

Tyrimo metu buvo suprojektuotas ir realizuotas prototipas, kuriame integruotas WireGuard VPT protokolas, TPM moduliu pagrįstas privačių raktų valdymas bei autentifikavimo mechanizmas, naudojantis TLS protokolą. Šis metodas užtikrina duomenų konfidencialumą, autentifikaciją ir vientisumą, neapkraunant sistemos pertekliniu skaičiavimu.

Eksperimentinės dalies metu atlikti interneto srauto pralaidumo ir saugumo testai leido palyginti siūlomo metodo efektyvumą su tradiciniais virtualaus privataus tinklo sprendimais, tokiais kaip OpenVPN ir IKEv2/IPsec. Tyrimo rezultatai parodė, kad siūlomas metodas pasižymi efektyviu našumo ir saugumo balansu, architektūriniu lankstumu bei tinkamumu naudoti realaus laiko transporto sistemose.

Obolevičius, Mantas. A Method for Encrypting Internet Traffic for Autonomous Vehicles. Master's Final Degree Project / supervisor prof. Algimantas Venčkauskas; Kaunas University of Technology. Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics Engineering (Computing).

Keywords: autonomous vehicles, encryption, virtual private network, WireGuard, TPM, TLS, secure communication, encrypted data channel, V2X communication, network security.

Kaunas, 2025. 65 p.

Summary

The main objective of this thesis is to develop and evaluate a secure internet traffic encryption method intended for communication between autonomous vehicles and roadside infrastructure units. The study analyzes modern network encryption technologies, virtual private network protocols, and their applicability in real-time transport systems, with particular attention paid to efficiency, security, and ease of implementation.

As part of the study, a prototype was designed and implemented, integrating the WireGuard VPN protocol, TPM-based private key management, and an authentication mechanism using the TLS protocol. This method ensures data confidentiality, authentication, and integrity without imposing excessive computational overhead on the system.

In the experimental part of the study, internet traffic throughput and security tests were conducted to compare the effectiveness of the proposed method with traditional virtual private network solutions such as OpenVPN and IKEv2/IPsec. The results showed that the proposed method offers an effective balance between performance and security, architectural flexibility, and suitability for use in real-time transport systems.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

AEAD – vienu metu šifruojantis ir tikrinantis duomenų autentiškumą algoritmas (angl. *Authenticated Encryption with Associated Data*).

CA – sertifikavimo įstaiga, išduodanti skaitmeninius sertifikatus (angl. *Certificate Authority*).

CAR-2-X – bendras autonominių transporto priemonių ryšio su kitais objektais terminas.

DDOS – paslaugų trikdymo ataka (angl. Distributed Denial of Service).

ECDSA – skaitmeninis parašas, pagrįstas elipsinėmis kreivėmis (angl. *Eliptic Curve Digital Signature Algorithm*).

GCM – autentifikuoto šifravimo režimas (angl. *Galois/Counter Mode*), naudojamas kartu su simetriniais šifrais.

GPS – globalinė padėties nustatymo sistema.

HKDF – HMAC pagrindu veikianti rakto išvesties funkcija (angl. *HMAC-based Key Derivation Function*), naudojama saugiam naujų kriptografinių raktų išvedimui iš pirminio bendro rakto.

HMAC – raktų pagrindu sukurtas pranešimų maišos kodas, naudojamas duomenų vientisumui ir autentiškumui užtikrinti (angl. *Hash-based Message Authentication Code*).

HTTP – hipertekstų persiuntimo protokolas (angl. *Hypertext Transfer Protocol*), naudojamas duomenų mainams tarp kliento ir serverio internete.

IKEv2/IPsec – saugus VPT ryšio protokolas, jungiantis greitą rakto apsikeitimą su saugaus IP srauto šifravimu (angl. *Internet Key Exchange version 2 / Internet Protocol Security*).

JSON – tekstinis duomenų mainų formatas, paremtas rakto–reikšmės poromis, dažnai naudojamas API komunikacijoje (angl. *JavaScript Object Notation*).

MANET – savaime organizuojamas mobilus tinklas be centrinės infrastruktūros (angl. *mobile ad-hoc network*).

MTU – didžiausias viename pakete galimas perduoti duomenų kiekis (angl. *Maximum transmision size*).

OpenVPN – atviro kodo virtualaus privataus tinklo sprendimas su SSL/TLS apsauga (angl. *Open-source Virtual Private Network solution*).

PKI – viešojo rakto infrastruktūra skirta saugiam šifravimui ir tapatybės patvirtinimui (angl. *Public Key Infrastructure*).

REST API – programavimo sąsaja, leidžianti skirtingoms programoms keistis duomenimis ar funkcionalumu (angl.. *Representational State Transfer Application Programming Interface*).

RSU – kelio infrastruktūros įrenginys, bendraujantis su transporto priemonėmis (angl. *Road-Side Units*).

SHA256 – kriptografinė maišos funkcija, užtikrinanti duomenų vientisumo tikrinimą

TCP / IP – pagrindinių interneto protokolų rinkinys, reglamentuojantis duomenų persiuntimą tarp įvairių tipų kompiuterių ir operacinių sistemų (angl. *transport control protocol*)

TCP protokolas – transporto lygmens protokolas, užtikrinantis patikimą duomenų perdavimą tarp įrenginių, naudojant klaidų aptikimą, pakartotinį siuntimą ir paketų eiliškumo atkūrimą.

TLS – transporto lygmens protokolas, skirtas saugumui užtikrinti TCP /IP tinkluose (angl. *Transport Layer Security*).

TPM – aparatinis įrangos saugos modulis, skirtas šifravimo raktų ir kitų jautrių duomenų apsaugai (angl. *Trusted Platform Module*).

UDP protokolas – duomenų perdavimo protokolas skirtas tiesiog siųsti duomenis be persiuntimo klaidų aptikimo ir taisymo.

V2I – ryšys tarp transporto priemonės ir infrastruktūros objektų (angl. *Vehicle-to-Infrastructure*).

V2V – tiesioginis ryšys tarp transporto priemonių (angl. *Vehicle-to-Vehicle).*

V2X – visapusis transporto priemonės ryšys su aplinka (angl. *Vehicle-to-everything*).

VANET – transporto priemonių tinklas, pagrįstas MANET principais (angl. *Vehicular Ad-hoc Network*).

VPT – virtualus privatus tinklas duomenų šifravimui.

Wi-Fi – belaidis vietinis tinklas.

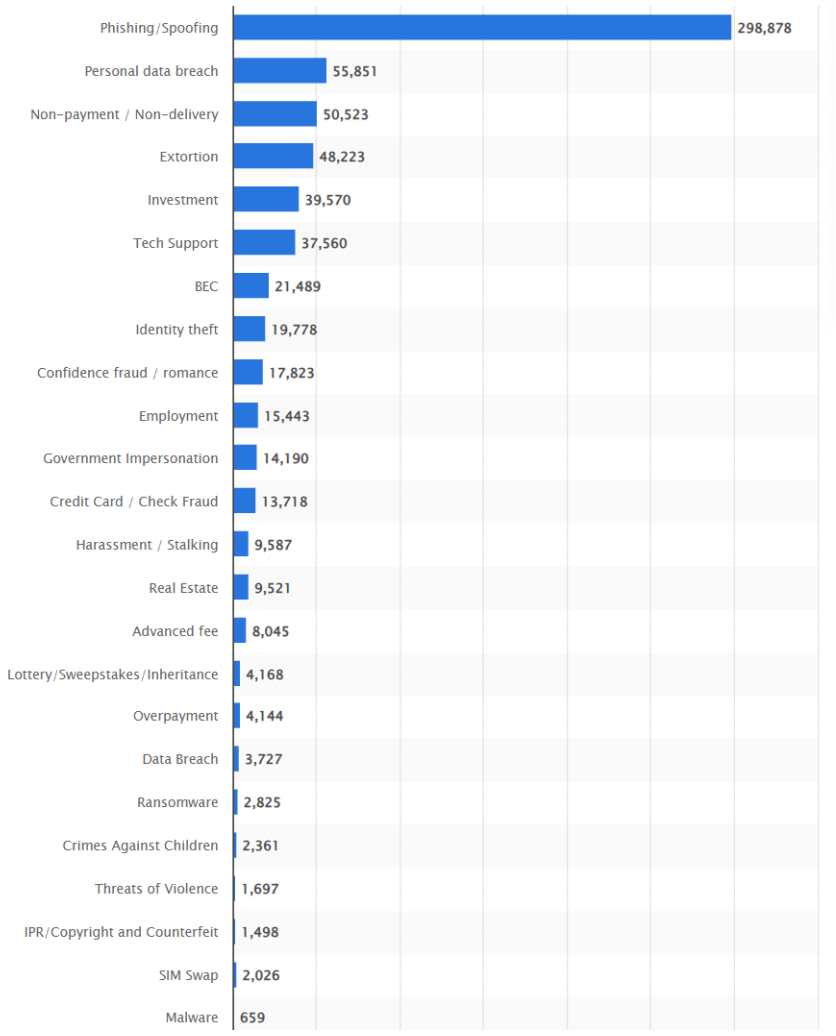
X.509 – tarptautinis standartas, apibrėžiantis skaitmeninių sertifikatų struktūrą, naudojamą tapatybės patvirtinimui ir viešųjų raktų sklaidai.

Įvadas

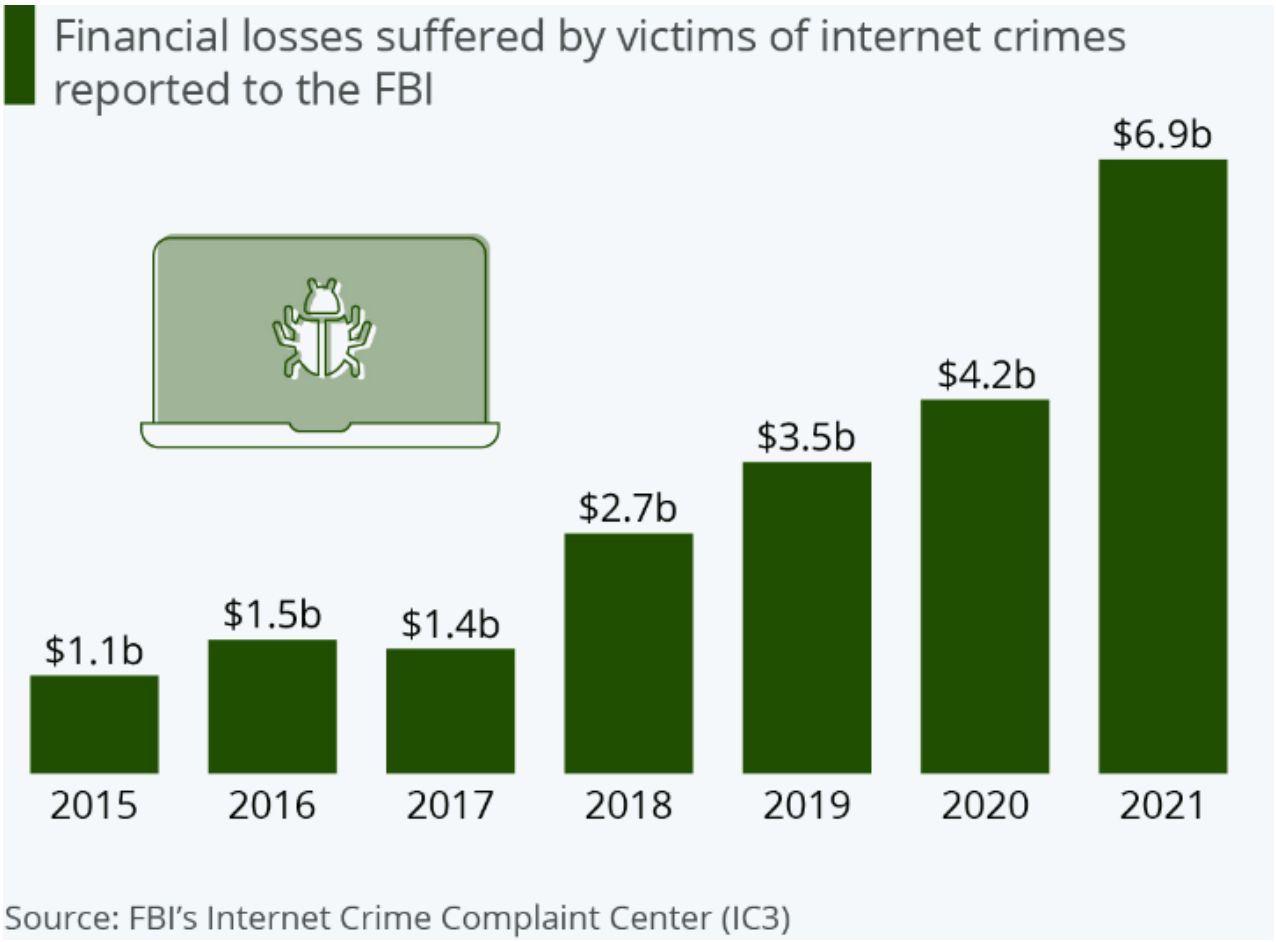
Autonominės transporto priemonės yra transporto priemonės, galinčios reaguoti į aplinką ir naviguoti be žmogaus įsikišimo. Autonominių transporto priemonių naudojamos technologijos apima įvairias sritis, įskaitant kompiuterių mokslą, robotiką ir mašininį mokymąsi. Remiantis „Statista“ duomenimis, tokių automobilių populiarumas pastaruoju metu ypač auga. Apskaičiuota, kad 2019 m. visame pasaulyje buvo parduota apie 1,4 mln. tokių transporto priemonių. Prognozuojama, kad 2030 m. pasaulyje bus parduota apie 58 mln. autonominių transporto priemonių [1].

Autonominių transporto priemonių korektiškas veikimas priklauso nuo įvairių mašinoje esančių daviklių rodmenų bei informacijos apsikeitimo greitaveikos. Tokie automobiliai yra prijungti prie interneto, o tai leidžia apsikeisti svarbia informacija tarpusavyje.

Savaime suprantama, tokios transporto priemonės potencialiai gali tapti kibernetinių nusikaltėlių taikiniu. Remiantis statistikos duomenimis, elektroninėje erdvėje nusikaltimo įvairovė plėtėsi (žr. 1 pav.) [2]. O ir pačių įsilaužimų skaičius didėjo remiantis „FBI Interneto nusikaltimų skundų centro“ informacija (žr. 2 pav.) [3], kurioje pateikiami finansiniai nuostoliai, susiję su kibernetinio saugumo incidentais.



1 pav. Kibernetinių nusikaltimų įvairovė [2]



2 pav. Kibernetinių įsilaužimų skaičius [3]

Kuriant autonomines transporto priemones, dėmesys į tokios kompleksiškos sistemos saugumą yra esminis klausimas. Dėl šių sistemų sudėtingumo ir sąveikos tarpusavyje jos yra pažeidžiamos įvairioms kibernetinėms atakoms, kurios gali pakenkti transporto priemonės ir jos keleivių saugai. Žemiau pateikiamos kelios potencialios tokių transportų priemonių kibernetinės grėsmės [4]:

1. **Atakos, orientuotos į transporto priemonės įrangą** apima telemetrijos kontrolerių realių parodymų iškraipymus, belaidžio Wi-Fi ryšio trikdymus, GPS rodmenų blokavimus, radarų atakas, kameros sensorių atakas, ultra garso sensorių atakas ir pan.
2. **Atakos, orientuotos į debesijos paslaugų tiekėjus** apima kelių transporto priemonių ataką, nulaužus vienos iš transporto priemonių kontrolerį.
3. **Paslaugos atsisakymo (angl. DDOS) ataka** apima transporto priemonės tinklo užtvindymą, kurio metu prarandamas ryšys su autonomine transporto priemone.
4. **Nuotolinis užgrobimas** apimą neteisėtą transporto priemonės užvaldymą per tinklo ryšį, pavyzdžiui, per internetą arba belaidžio ryšio sistemą. Užpuolikai gali pasinaudoti transporto priemonės programinės įrangos ar tinklo infrastruktūros pažeidžiamumais, siekdami įgyti prieigą prie jos valdymo sistemų. Tokiu atveju tampa įmanoma perimti transporto priemonės judėjimo, navigacijos ar kitų svarbių funkcijų kontrolę.

Darbo tikslas – sukurti efektyvų ir patikimą informacijos apsikeitimo tarp autonominių transporto priemonių metodą.

Darbo uždaviniai:

1. atlikti autonominių transporto priemonių tarpusavio komunikacijos ir saugumo reikalavimų analizę;
2. išanalizuoti ir atrinkti tinkamus saugaus duomenų perdavimo bei programinės įrangos integralumo tikrinimo metodus;
3. sukurti informacijos apsikeitimo algoritmą, taikant pasirinktus informacijos saugumo sprendimus;
4. eksperimentiškai įvertinti realizuoto sprendimo našumą ir tinkamumą autonominių sistemų aplinkoje.

1. Autonominių transporto priemonių tinklų saugumo analizė

1.1. Autonominių transporto priemonių tinklo architektūros ypatumai

Autonominės transporto priemonės veikia specializuotuose tinkluose – VANET (angl. *Vehicular Ad-hoc Network*) arba CAR-2-X, kurie yra mobiliojo ryšio MANET (angl. *mobile ad-hoc network*) potipiai. Šie tinklai skirti ryšiui palaikyti tarp transporto priemonių bei kelio infrastruktūros komponentų, tokių kaip RSU (angl. *Road-Side Units*). Tokie tinklai taikomi įvairiose srityse, pavyzdžiui, eismui valdyti, susidūrimų prevencijai užtikrinti, informacinėms-pramoginėms paslaugoms suteikti (angl. *infotainment*) [5].

Vienas iš pagrindinių šių tinklų iššūkių yra tinkamai veikiančios infrastruktūros trūkumas. Skirtingai nei tradiciniuose laidiniuose tinkluose, VANET ir CAR-2-X formuojami dinamiškai pačių transporto priemonių, kurios nuolat keičia padėtį. Ši savybė apsunkina pastovaus ryšio palaikymą, ypač didelio tankumo teritorijose, pavyzdžiui, miestuose.

1.2. Transporto priemonių tinklo architektūros modelis

Šiame poskyryje pateikiama VANET ir Europoje taikomų CAR-2-X tinklų architektūrų analizė.

1.2.1. VANET tinklo domenai

VANET tinklo architektūros komponentai yra skirstomi į atskirus domenus. Tokių domenų yra trys [5]:

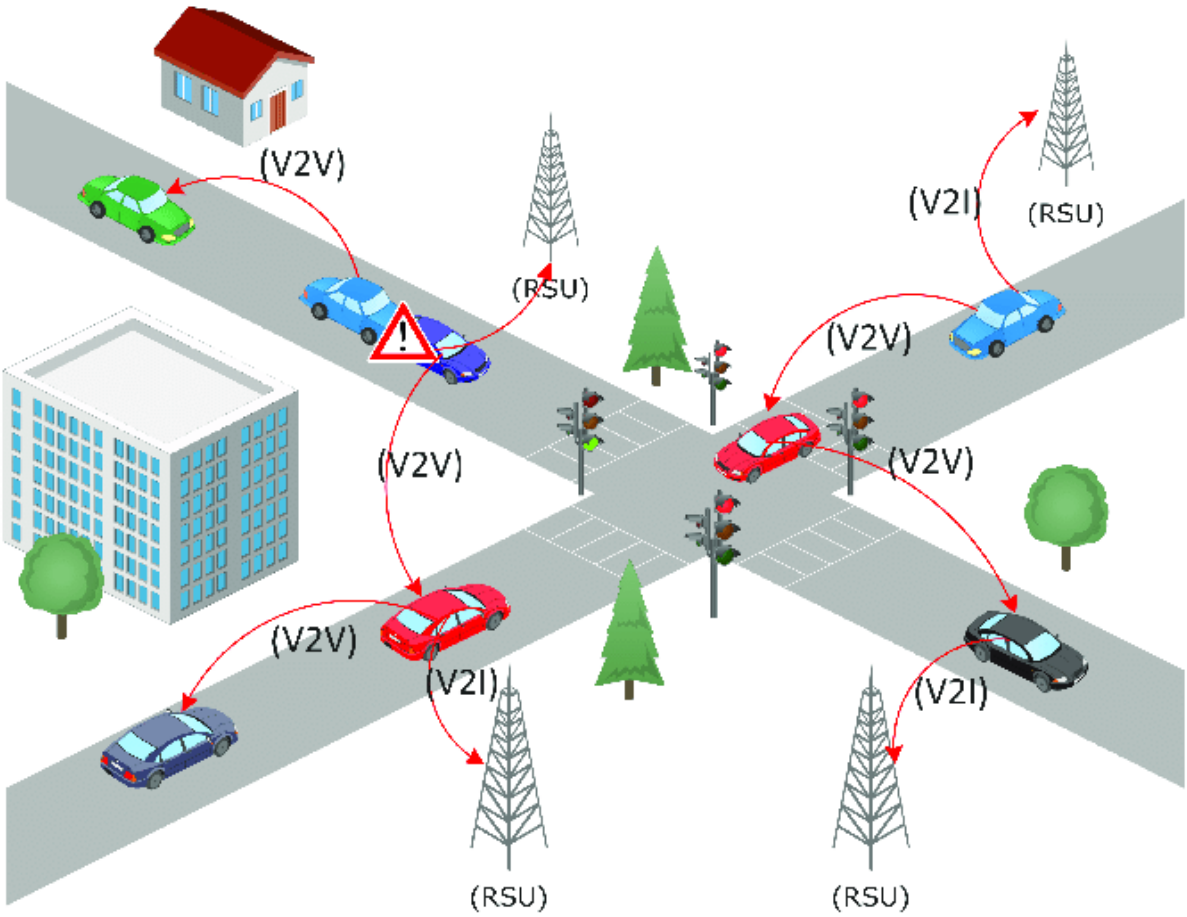
1. **Mobilusis domenas** apima transporto priemonę ir mobiliuosius įrenginius. Pačios transporto priemonės tipas gali būti įvairus: automobiliai, traukiniai, autobusai. Mobiliaisiais įrenginiais gali būti išmanieji telefonai, kompiuteriai, išmanieji laikrodžiai ir pan.
2. **Infrastruktūros domenas** apima kelių infrastruktūros objektus: šviesoforus, kameras, eismo valdymo centrus ir pan.
3. **Bendrasis domenas** apima interneto ryšį bei privačias paslaugų platformas.

Pateikiama VANET tinklo architektūros diagrama (žr. 3 pav.). Joje vaizduojamos kelios transporto priemonės, besikeičiančios informacija tarpusavyje per tiesioginius ryšius **V2V** (angl. *Vehicle-to-Vehicle*) bei sąveikaujančios su kelių infrastruktūros objektais per **V2I** (angl. *Vehicle-to-Infrastructure*) ryšį, pastarajam naudojant **RSU**.

Keletas schemoje vaizduojamų situacijų pavyzdžių:

- transporto priemonės tarpusavyje keičiasi įspėjimais apie kliūtis kelyje;
- iš RSU gaunama informacija apie šviesoforų būsenas;
- vykdomas koordinuotas transporto priemonių judėjimas per sankryžas;
- automobilių trajektorijos koreguojamos atsižvelgiant į kitų eismo dalyvių veiksmus.

Diagramoje aiškiai matyti, kaip **mobilusis domenas** (transporto priemonės) sąveikauja su **infrastruktūros domenu** (šviesoforai, RSU), o **bendrasis domenas** atvaizduojamas kaip ryšio perdavimo fonas, užtikrinantis platesnį duomenų perdavimo tinklą.



3 pav. VANET tinklo architektūros schema [6]

1.2.2. CAR-2-X tinklo domenai

VANET tinklai plačiau taikomi Jungtinėse Amerikos Valstijose, o Europoje transporto priemonių tarpusavio komunikacija remiasi CAR-2-X modeliu, kurį vysto CAR-2-CAR (C2C) komunikacijos konsorciumas[1]. Tai Europos mokslinių tyrimų ir plėtros iniciatyva, kurios tikslas yra skatinti belaidžio ryšio technologijų diegimą transporto sistemose. Projektas orientuotas į vieningos komunikacijos platformos (angl. *Cooperative Awareness Communication*) kūrimą.

CAR-2-X architektūrą sudaro [5]:

1. **Transporto priemonės domenas** apima vieną arba kelis taikomųjų funkcijų įrenginius (AU, angl. *Application Unit*) bei transporto priemonės borto įrenginį (OBU, angl. *On-Board Unit*). AU gali būti integruotas į automobilį arba veikti kaip atskiras nešiojamasis įrenginys, pavyzdžiui, išmanusis telefonas ar nešiojamasis kompiuteris. Šie įrenginiai naudoja OBU teikiamas ryšio galimybės. Abu prietaisai gali būti sujungti tiek laidiniu, tiek belaidžiu ryšiu.

[1]CAR-2-CAR komunikacijos konsorciumas: https://www.car-2-car.org/

2. **Specialusis domenas** apima OBU ir RSU įrenginius, įdiegtus tam tikrose kelio vietose. OBU gali bendrauti tiek tiesiogiai, tiek per kelis tinklo mazgų šuolius, naudojant trumpojo nuotolio belaidžio ryšio technologijas.
3. **Infrastruktūros domenas** apima RSU įrenginius bei interneto prieigos taškus. Jeigu šie taškai negali užtikrinti prieigos prie interneto, komunikacijai gali būti pasitelkiamos mobiliojo ryšio technologijos, tokios kaip 4G arba 5G.

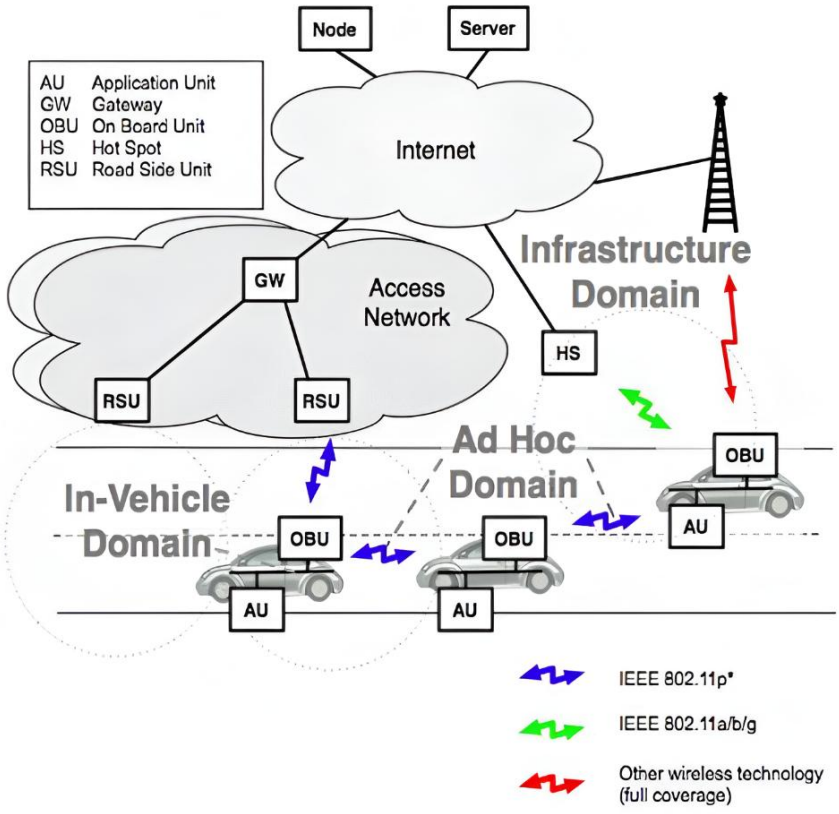
Žemiau pateikiama CAR-2-X tinklo architektūros schema (žr. 4 pav.). Joje pavaizduotos pagrindinės tinklo sritys, jų veikimo principai ir tarpusavio ryšiui naudojamos technologijos.

Transporto priemonės srityje veikia taikomųjų funkcijų įrenginiai (AU) ir borto įrenginiai (OBU), kurie tarpusavyje keičiasi duomenimis automobilio viduje, naudodami laidinį arba belaidį ryšį.

Tiesioginio ryšio segmente, kuriame komunikacija vyksta tarp transporto priemonių borto įrenginių (OBU–OBU), taikoma IEEE 802.11p technologija (mėlyna spalva), leidžianti pasiekti mažą perdavimo delsą tarp judančių mazgų.

Infrastruktūros srityje funkcionuoja RSU bei kiti tinklo komponentai, tokie kaip prieigos taškai, kurie, pasitelkiant prieigos tinklą ir maršruto valdymo elementus (tinklo šliuzus), sujungiami su internetu. Šioje srityje naudojamos įvairios belaidžio ryšio technologijos: *IEEE 802.11a/b/g* (žalia spalva) bei mobiliojo ryšio sprendimai (raudona spalva). Taip užtikrinamas globalus tinklo pasiekiamumas.

Schema taip pat iliustruoja duomenų srautų kelią tarp skirtingų ryšio sluoksnių, pavyzdžiui, kaip informacija perduodama iš transporto priemonės į centrinį serverį per kelias tinklo jungtis.



4 pav. C2C tinklo architektūros diagrama [5]

1.3. Komunikacijos architektūra

Remiantis pateikta VANET ir CAR-2-X tinklų architektūrų analize, šiame poskyryje detaliau nagrinėjami autonominių transporto priemonių tarpusavio komunikacijos principai, susiję su realiomis eksploatacinėmis sąlygomis ir tinklo veikimo aspektais.

1.3.1. Komunikacijos architektūros aspektai

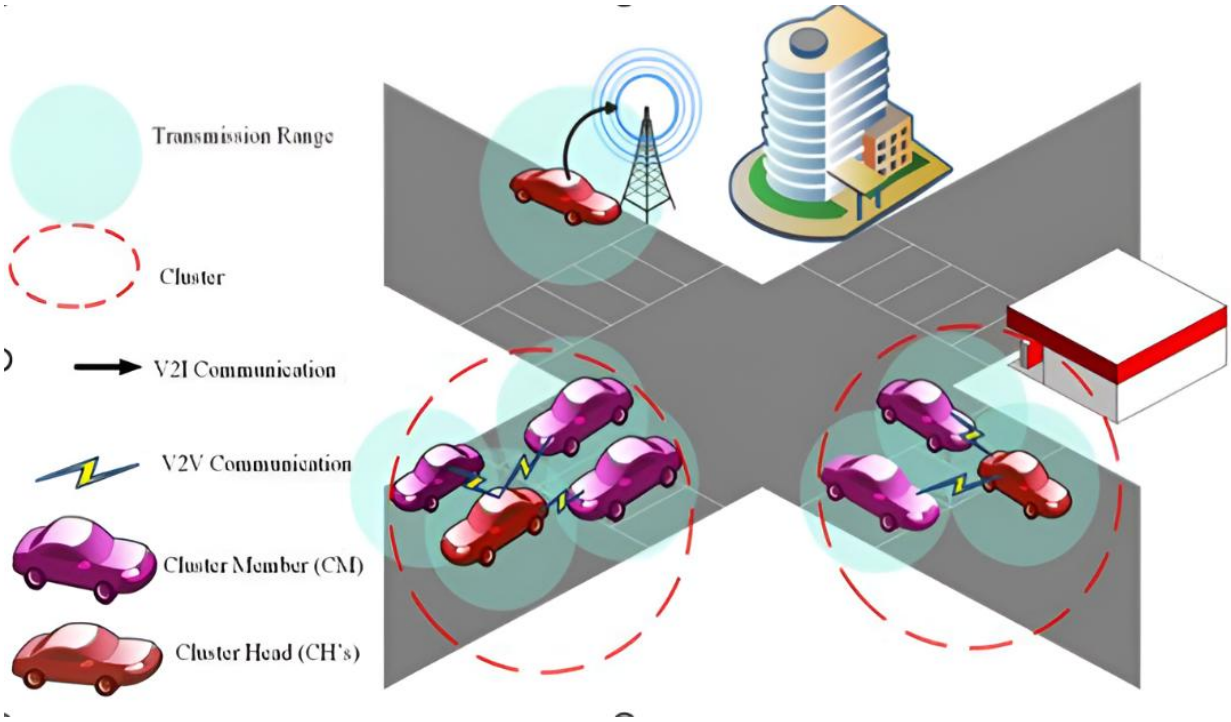
Analizuojant komunikacijos architektūrą, išskiriami keli svarbūs veiksniai, darantys didžiausią įtaką tinklo efektyvumui [5]:

- **Judėjimo modelis.** Autonominių transporto priemonių tinklai dažniausiai yra organizuojami remiantis kelių topologiją. Paprastai išskiriamos trys tokios topologijos: miesto, kaimo ir greitkelių. Miesto vietovėse, kur transporto priemonių tankis didesnis, būtinas didesnis kelio infrastruktūros objektų (RSU) kiekis, norint palaikyti stabilų ir patikimą ryšį.
- **Eismo intensyvumas.** Transporto priemonių komunikacijos protokolai turi būti jautrūs eismo pokyčiams ir į juos reaguoti. Kaimo vietovėse, kuriose transporto priemonių skaičius mažas, svarbu užtikrinti nepertraukiamą ryšį net ir retai pasitaikant tinklo mazgams. Gali būti, kad tinkle tam tikrą laiką nebus nė vienos transporto priemonės. Be to, pati transporto priemonė turi gebėti greitai prisijungti prie skirtingų prieigos tinklų.
- **Heterogeniškumas.** Tinklo mazgai dažnai skiriasi savo charakteristikomis ir funkcionalumu. Transporto priemonės yra mobilūs mazgai, pasižymintys skirtingais moduliais, ryšio technologijomis bei paskirtimi. Kelio infrastruktūros objektai (RSU) yra stacionarūs mazgai, įrengti strategiškai tam tikrose vietose ir palaikantys visą tinklo funkcionalumą.

Atsižvelgiant į tai, kad autonominių transporto priemonių tinklo mazgai dažnai keičia savo poziciją, tampa sudėtinga palaikyti patikimą ir stabilų ryšį. Dėl greito ir atsitiktinio tinklo mazgų (angl. *network nodes*) judėjimo kyla prieinamumo, plečiamumo (angl. *scalability*) ir tinklo stabilumo problemos, kurios daro tiesioginę įtaką paslaugų kokybei [7].

Vienas iš galimų šių problemų sprendimo būdų yra tinklo grupavimas [7]. Tokiu atveju kelios transporto priemonės sudaro lokalų tinklą, kuris veikia kaip atskiras, savarankiškas vienetas. Pavyzdžiui, keturios transporto priemonės gali sudaryti atskirą ryšio grupę (žr. 5 pav.).

Schemoje pavaizduota autonominių transporto priemonių komunikacijos sistema, paremta tinklo grupavimo principu. Violetinės spalvos transporto priemonės žymi grupės narius (CM, angl. *Cluster Members*), o raudonos spalvos – grupės vadus (CH, angl. *Cluster Heads*), koordinuojančius informacijos apsikeitimą. Kiekvienos grupės aprėptis pažymėta raudonu brūkšniuotu apskritimu, nurodančiu klasterio ribas, o šviesiai mėlynos srities žiedai žymi ryšio aprėpties zonas (angl. *Transmission Range*). Žaibo formos linijos vaizduoja V2V (angl. *Vehicle-to-Vehicle*) ryšį tarp transporto priemonių, o juodos rodyklės – V2I (angl. *Vehicle-to-Infrastructure*) ryšį tarp transporto priemonių ir kelio infrastruktūros objektų. Tokia architektūra leidžia efektyviai valdyti duomenų srautus bei palaikyti patikimą ryšį esant dideliam tinklo mazgų mobilumui.



5 pav. Transporto priemonių grupavimas į atskirus tinklus [7]

1.4. Saugų ryšį užtikrinantys sprendimai autonominių transporto priemonių tinkluose

Šiame poskyryje analizuojami pagrindiniai kriptografiniai sprendimai, taikytini siekiant užtikrinti saugų duomenų perdavimą tarp autonominių transporto priemonių ir kelio infrastruktūros elementų. Atsižvelgiant į transporto sistemos mobilumą bei realaus laiko veikimo reikalavimus, saugos priemonės turi būti efektyvios išteklių naudojimo požiūriu, greitos, patikimos ir lengvai pritaikomos dinamiškoje tinklo topologijoje.

1.4.1. V2X saugumo architektūra

Standartinė V2X (angl. *Vehicle-to-Everything*) architektūra remiasi viešųjų raktų infrastruktūra (angl. *Public Key Infrastructure*, PKI), kurioje kiekvienai transporto priemonei išduodamos laikinosios skaitmeninės tapatybės (angl. *pseudonymous identities*), patvirtintos sertifikavimo centro (angl. *Certificate Authority*). Žinučių pasirašymui naudojamas ECDSA (angl. *Elliptic Curve Digital Signature Algorithm*) algoritmas, leidžiantis transporto priemonėms autentifikuoti siunčiamas žinutes ir užtikrinti jų vientisumą. Tokiu būdu tiek transporto priemonių tarpusavio (V2V), tiek ryšio su infrastruktūra (V2I) metu perduodami duomenys gali būti patikimai patvirtinti, o jų kilmė – atsekama.[8].

ECDSA yra elipsinių kreivių kriptografija paremtas skaitmeninio parašo algoritmas, naudojamas autentifikacijai ir duomenų integralumo užtikrinimui. Dėl savo efektyvumo ir saugumo ECDSA yra vienas iš pagrindinių algoritmų, apibrėžtų V2X komunikacijos saugumo standartuose, įskaitant *IEEE 1609.2* [9].

ECSA leidžia pasirašyti kiekvieną siunčiamą žinutę privačiu raktu, o gavėjas, naudodamas atitinkamą viešąjį raktą, gali patikrinti parašo galiojimą. Tokiu būdu užtikrinamas žinutės autentiškumas ir

integralumas, net jei duomenų perdavimo kanalas nėra šifruotas. Žinučių pasirašymas pagrįstas elipsinių kreivių diskrečiųjų logaritmų problema (angl. *Elliptic Curve Discrete Logarithm Problem*, ECDLP), kuri laikoma skaičiavimo požiūriu sudėtinga.

Tipiška ECDSA parašo struktūra susideda iš reikšmių poros (*r, s*), kurios apskaičiuojamos naudojant atsitiktinę reikšmę *k*, elipsinės kreivės bazinį tašką *G,* pranešimo maišos funkciją *H(m)* bei siuntėjo privatų raktą *d*. Pasirašymo metu sukuriamas taškas $R = k \cdot G$, iš kurio paimama X koordinatė *r,* o reikšmė *s* apskaičiuojama kaip:

$$s = k^{-1} \cdot (H(m) + d \cdot r) \bmod n$$

Kur:

- *d* – siuntėjo privatusis raktas;
- *n* – kreivės taškų grupės tvarka.

Gavėjas, naudodamas siuntėjo viešąjį raktą $Q = d \cdot G$, gali patikrinti parašą, atkurdamas atitinkamą tašką elipsinėje kreivėje ir įsitikindamas, kad jis atitinka pranešimo maišos ir parašo komponentų reikšmes. Tikrinimo formulė leidžia nustatyti, ar (*r, s*) iš tikrųjų buvo sugeneruota žinant atitinkamą privatų raktą ir tą konkretų pranešimą.

Nepaisant aukšto saugumo lygio, ECDSA reikalauja papildomų resursų tiek parašo generavimui, tiek tikrinimui, taip pat priklauso nuo gerai suvaldytos viešųjų raktų infrastruktūros (PKI), kuri apima sertifikatų, pseudoniminių tapatybių ir jų galiojimo valdymo sistemą.

1.4.2. ChaCha20 algoritmas

ChaCha20 yra srautinis šifravimo algoritmas, plačiai naudojamas moderniose saugumo sistemose, įskaitant virtualaus privataus tinklo protokolus, tokius kaip WireGuard.

Skirtingai nei tradicinis AES algoritmas, kuris veikia bloko pagrindu ir reikalauja specializuotos aparatinės įrangos optimizacijai (pvz., AES-NI instrukcijų), ChaCha20 pasižymi puikiu našumu net procesoriuose, neturinčiuose šių spartinimo priemonių. Tai ypač aktualu autonominėse transporto priemonėse, kuriose naudojama įvairaus pajėgumo įterptinė įranga [10]. Šis algoritmas taiko tris operacijas: sveikųjų skaičių sudėtį, išimtinė disjunkciją (XOR) bei bitų rotaciją (angl. *bit rotation*).

Be to, algoritmas efektyviai šifruoja nedidelės apimties duomenis, o tai būdinga dažnam ir trumpam autonominių transporto priemonių komunikavimui (pvz., koordinatėms, greičiui, įvykių įrašams perduoti). Dėl šių priežasčių ChaCha20 dažnai laikomas vienu tinkamiausių pasirinkimų realiojo laiko ryšio šifravimui mobiliuose įrenginiuose [10].

1. Pradiniai algoritmo parametrai [10]:
    - 256 bitų šifravimo raktas;
    - 32 bitų pradinis skaitiklis (angl. *Counter*). Tai gali būti bet koks skaičius, bet paprastai yra nulis arba vienas. Tikslinga naudoti vieną, jei nulinį bloką naudojame kažkam kitam, pavyzdžiui, generuojame vienkartinį autentifikavimo raktą;

- 96 bitų vienkartinė reikšmė (angl. *Nonce*), skirta kiekvienos šifravimo operacijos unikalumui užtikrinti;
- Paprastas tekstas (angl. *Plaintext*), kuris bus šifruojamas;
- Papildomi duomenys (angl. *associated data*), kurie nėra šifruojami, bet turi būti autentifikuoti kartu.

2. Veikimo principas [10]:
    a. Pradžioje ChaCha20 algoritmas suformuoja **512 bitų vidinę būseną** iš konstantų, rakto, skaitiklio ir vienkartinės reikšmės.
    b. Ši būsena toliau apdorojama atliekant 20 iteracijų, per kurias taikoma vadinamoji „*quarter-round*“ funkcija kuri maišo po keturis 32 bitų dydžio duomenų žodžius, kad pasiektų tolygų informacijos pasklidimo (difuzijos) lygį visoje būsenoje.
    c. Po visų iteracijų generuojamas 64 baitų ilgio raktų srautas, kuris naudojamas šifravimui.
    d. Kiekvienas paprastojo teksto blokas šifruojamas taikant išimtinės disjunkcijos (XOR) operaciją su sugeneruotu raktų srauto bloku, vadovaujantis formule: $\text{šifruotas\_blokas} = \text{tekstas} \oplus \text{raktų\_srautas}$
    e. Jei naudojamas *Poly1305* algoritmas, jis sugeneruoja autentifikavimo kodą (MAC), kuris leidžia gavėjui įsitikinti, kad gauti duomenys nebuvo pakeisti.

Žemiau pateikiamas šio algoritmo šifravimo operacijos pseudokodas (žr. 6 pav.):

```
chacha20_encrypt(key, counter, nonce, plaintext):

    for j = 0 to floor(len(plaintext) / 64) - 1:

        key_stream = chacha20_block(key, counter + j, nonce)

        block = plaintext[(j * 64)..(j * 64 + 63)]

        encrypted_message += block ^ key_stream

    end


    if (len(plaintext) % 64) != 0:

        j = floor(len(plaintext) / 64)

        key_stream = chacha20_block(key, counter + j, nonce)

        block = plaintext[(j * 64)..len(plaintext) - 1]

        encrypted_message += (block ^ key_stream)[0..len(plaintext) % 64]

    end


    return encrypted_message

end
```

6 pav. ChaCha20 šifravimo algoritmo pseudokodas [10]

ChaCha20 algoritmas veikia AEAD režimu (angl. *Authenticated Encryption with Associated Data*). Tai šifravimo schema, leidžianti vienu metu šifruoti duomenis ir apskaičiuoti žymą, apimančią tiek šifruotą turinį, tiek papildomus nešifruojamus duomenis (pvz., antraštes). Tai užtikrina, kad bet kokie paketo turinio ar struktūros pakeitimai būtų nedelsiant aptinkami [10].

Poly1305 yra vienkartinio naudojimo pranešimų autentifikavimo kodas, užtikrinantis duomenų vientisumą ir autentiškumą [10]. Algoritmas veikia kaip universali maišos funkcija, kuriai taikomas modulis $p = 2^{130} - 5$, o rezultatas yra 128 bitų autentifikavimo žyma. Jo saugumas grindžiamas vienkartinio 256 bitų ilgio rakto naudojimu, kuris dalijamas į dvi 128 bitų reikšmes: *r* – naudojama kaip daugiklis maišos funkcijoje, ir *s* – pridedama prie rezultatų prieš išvedant autentifikavimo žymą.

Algoritmo veikimo eiga yra paremta tarpinės sumos atnaujinimu, kiekvienam pranešimo blokui atliekant aritmetinius veiksmus. Pirmiausia, pranešimas padalijamas į 16 baitų ilgio blokus. Jei paskutinis blokas yra trumpesnis, jis užpildomas nuliais. Kiekvienas blokas interpretuojamas kaip mažo reikšmingumo baitų (angl. *little-endian*) sveikasis skaičius, prie kurio pridedamas papildomas bitas, siekiant išlaikyti vientisą formatą.

Kiekvieno bloko reikšmė sudedama su esama tarpinių skaičiavimų suma, o rezultatas padauginamas iš *r*. Tuomet apskaičiuojama liekana, dalijant iš pirminio skaičiaus *p*. Pasibaigus visų blokų apdorojimui, prie tarpinių rezultatų pridedama reikšmės *s* vertė, o gauta suma sumažinama iki 128 bitų pločio. Ši reikšmė tampa galutine autentifikavimo žyma, naudojama pranešimo vientisumui ir autentiškumui patikrinti. Šio algoritmo pseudokodas pateikiamas žemiau (žr. 7 pav.).

```
clamp(r): r &= 0x0ffffffc0ffffffc0ffffffc0fffffff

    poly1305_mac(msg, key):

      r = (le_bytes_to_num(key[0..15])

      clamp(r)

      s = le_num(key[16..31])

      accumulator = 0

      p = (1<<130)-5

      for i=1 upto ceil(msg length in bytes / 16)

        n = le_bytes_to_num(msg[((i-1)*16)..(i*16)] | [0x01])

        a += n

        a = (r * a) % p

        end

      a += s

      return num_to_16_le_bytes(a)

      end
```

7 pav. Poly1305 pseudokodas [10]

Šis algoritmas yra integruojamas į WireGuard protokolą [10]. Skiriasi tai, kad reikšmės *r* ir *s* jame nėra parenkamos tiesiogiai, o išvedamos naudojant ChaCha20 blokinę funkciją. Naudojant skaitiklį, su reikšme 0 (angl. *counter = 0*) ir vienkartinę reikšmę, ChaCha20 sugeneruoja 512 bitų išvestį, iš kurios pirmieji 256 bitai naudojami kaip vienkartinis raktas: pirmieji 128 bitai priskiriami *r*, o likusieji *s*. Reikšmė *r* prieš naudojimą modifikuojama pagal bitų apribojimus (angl. *clamping*), siekiant užtikrinti saugumą.

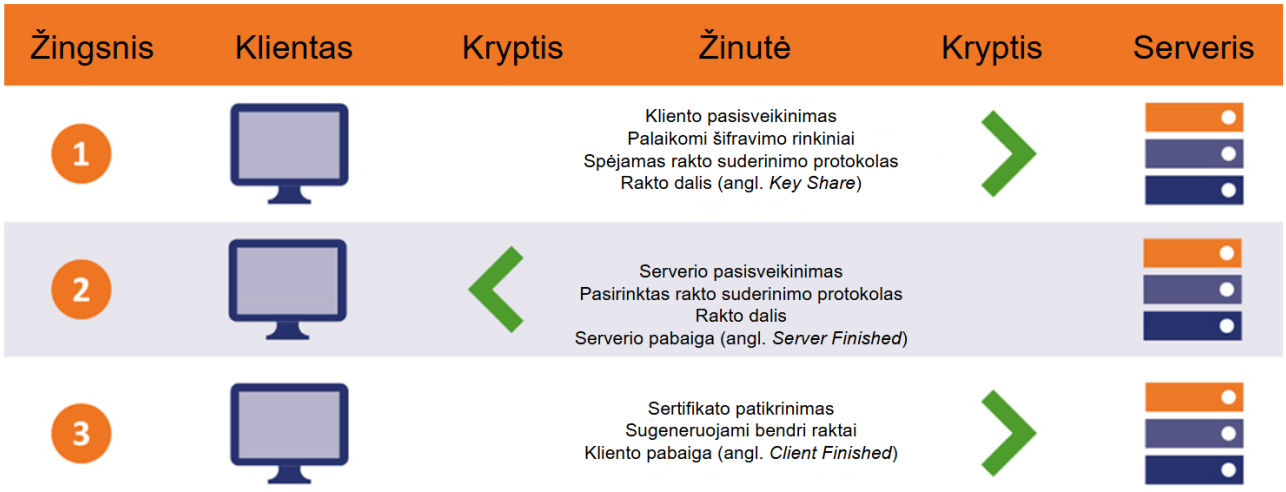
Taip suformuotas raktas leidžia Poly1305 algoritmui apskaičiuoti autentifikavimo žymą kiekvienam WireGuard protokolo paketo šifruotam turiniui bei papildomiems duomenims. Šis procesas leidžia vienu metu užtikrinti tiek konfidencialumą, tiek duomenų autentiškumą.

1.4.3. TLS protokolas

TLS (angl. *Transport Layer Security*) – tai plačiai naudojamas kriptografinis protokolas, skirtas užtikrinti saugų duomenų perdavimą TCP/IP tinkluose. Jis užtikrina trijų pagrindinių saugumo savybių: konfidencialumo, vientisumo ir autentiškumo įgyvendinimą. TLS veikia transporto lygmenyje ir dažniausiai taikomas apsaugant žiniatinklio srautą, elektroninį paštą, VPT tunelius bei kitus tinklo ryšius.

TLS naudoja hibridinį šifravimo metodą: ryšio pradžioje vykdomas asimetrinis raktų apsikeitimas, po kurio pereinama prie simetrinio šifravimo, naudojant derybų metu sugeneruotą sesijos raktą. Duomenų vientisumas užtikrinamas taikant pranešimų autentifikavimo kodų algoritmus, o šiuolaikinėse TLS 1.2 ir TLS 1.3 versijose įdiegtos pažangios AEAD schemos (pvz., AES-GCM, ChaCha20-Poly1305), leidžiančios vienu metu užtikrinti tiek šifravimą, tiek autentifikavimą.

Žemiau pateikiama supaprastinta TLS 1.3 protokolo autentifikacijos schema (žr. 8 pav.), kurioje vaizduojami pagrindiniai kliento ir serverio keitimosi pranešimais žingsniai, reikalingi saugiam ryšiui užmegzti.



8 pav. TLS 1.3 autentifikacija [11]

1.4.4. WireGuard protokolas

WireGuard yra modernus, atvirojo kodo, didelio našumo virtualaus privataus tinklo (angl. *Virtual Private Network*) protokolas, skirtas užtikrinti saugų ryšį tarp dviejų ar daugiau tinklo mazgų. Jis išsiskiria paprastumu, efektyvumu ir naujausių kriptografinių sprendimų taikymu. Dėl mažo kodo eilučių kiekio ir aiškios struktūros, WireGuard laikomas lengvai integruojamu ir audituojamu sprendimu tiek vartotojams, tiek saugumo specialistams [12].

Protokolas veikia tinklo sluoksnyje (OSI 3 lygmuo) ir naudoja šiuolaikinius kriptografinius algoritmus, tokius kaip:

- Elipsinę kreivę Curve25519 raktų mainams;
- ChaCha2*0* šifravimo algoritmą;
- Poly1305 autentifikavimui;
- BLAKE2s maišos funkcijoms;
- HDKF (angl. *HMAC-based Extract-and-Expand Key Derivation Function*) raktų išvedimo funkcija. Naudojama norint išgauti vieną ar kelis kriptografinius raktus iš bendros slaptos reikšmės.

Skirtingai nei tokie tradiciniai virtualus privataus tinklo sprendimai kaip OpenVPN ar IPsec, kurie dažnai reikalauja sudėtingų konfigūracijų, sertifikatų infrastruktūros ar trečiųjų šalių valdymo įrankių, WireGuard remiasi itin paprastu identifikavimo modeliu. Kiekvienas tinklo mazgas (angl. *peer*) identifikuojamas vien tik pagal savo viešąjį raktą, kuris yra 32 baitų ilgio elipsinės kreivės Curve25519 taškas. Kartu su raktu kiekvienam tinklo mazgui priskiriamas fiksuotas IP adresas, kuris leidžia aiškiai apibrėžti, kokiu adresu galima siųsti šifruotą srautą.

Šis mechanizmas leidžia išvengti sudėtingo sertifikatų bei dinaminio rakto valdymo, kuris dažnai būdingas senesniems virtualaus privataus tinklo protokolams. Dėl šios priežasties WireGuard diegimas yra paprastas, greitas ir tinkamas tokioms dinamiškoms aplinkoms kaip autonominių transporto priemonių tinklai. Vis dėlto, jeigu to reikalauja saugumo architektūra, prie WireGuard taip pat gali būti pritaikyta papildoma autentifikacija, pavyzdžiui, naudojant skaitmeninius sertifikatus ar papildomus kontrolinius mechanizmus per išorinę programinę įrangą ar API (angl. *Application Programming Interface*) sprendimus.

Šio virtualaus privataus tinklo protokolo veikimo principas grindžiamas nuosekliais komunikacijos žingsniais:

1. **Paketo priėmimas**. Duomenų paketas pirmiausia pasiekia WireGuard sąsają (*interface*), veikiančią kaip virtualus tinklo prievadas.
2. **Sesijos nustatymas.** Naudodamas paketo antraštę, protokolas patikrina ar paketas susijęs su jau egzistuojančia tinklo mazgo sesija. Taip pat patikrinamas pranešimo skaitiklis, siekiant išvengti pakartojimo atakų (angl. *replay attacks*). Jei autentifikacija naudojant simetrinį raktą nepavyksta, paketas atmetamas.
3. **IP adreso atnaujinimas.** Jeigu autentifikacija yra sėkminga, iš UDP/IP antraštės paimtas šaltinio IP adresas naudojamas tinklo mazgo galutinio taško atnaujinimui (angl. *endpoint*), kuris nusako iš kur ateina šifruotas srautas.

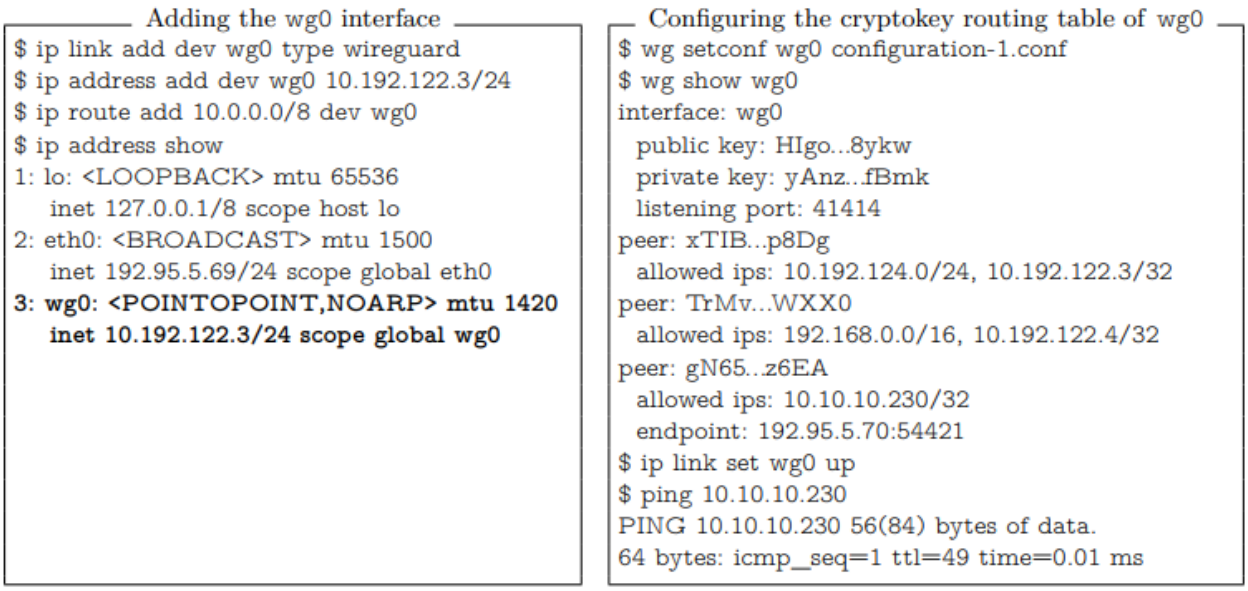
4. **Paketo turinio šifravimas.** Šifruotas paketo turinys yra iššifruojamas. Jeigu paketas nėra IP tipo, jis atmetamas. Priešingu atveju tikrinama, ar iššifruoto paketo šaltinio IP adresas turi atitikmenį WireGuard kriptografinių raktų maršrutų lentelėje. Pavyzdžiui, jei IP adresas atitinka kripto raktui priskirtą leistiną adresų diapazoną (pvz., 192.168.31.28), paketas priimamas. Jei ne (pavyzdžiui, gautas IP adresas yra 10.192.122.3), jis atmetamas.
5. **Paketo perdavimas.** Jei visi saugumo ir maršruto tikrinimai sėkmingi, paketas įterpiamas į WireGuard virtualią sąsają ir perduodamas operacinei sistemai tarsi jis būtų gautas iš fizinio tinklo prievado.

Žemiau pateikiama iliustracija (žr. 9 pav.), vaizduojanti WireGuard protokolo veikimą praktikoje, pradedant nuo sąsajos (wg0) sukūrimo iki saugaus ryšio užmezgimo tarp dviejų tinklo mazgų. Kairėje pusėje pateikiama komandų seka, kaip sistemoje sukuriama wg0 sąsaja, jai priskiriamas IP adresas ir maršrutas. Tai atspindi pirmuosius ryšio užmezgimo veiksmus, kai paketas patenka į WireGuard sąsają.

Dešinėje pusėje pateiktas kriptografinių raktų ir maršrutų lentelės (angl. *cryptokey routing table*) konfigūravimas, kuriame aiškiai matoma ši informacija:

1. kiekvieno tinklo mazgo viešieji raktai;
2. leidžiami IP adresai;
3. galutiniai taškai, nurodantys, kur reikia siųsti paketus.

Galiausiai komanda *ping* parodo sėkmingą duomenų siuntimą per virtualų tunelį. Tai reiškia, kad autentifikavimo ir maršrutizavimo procesas suveikė teisingai.

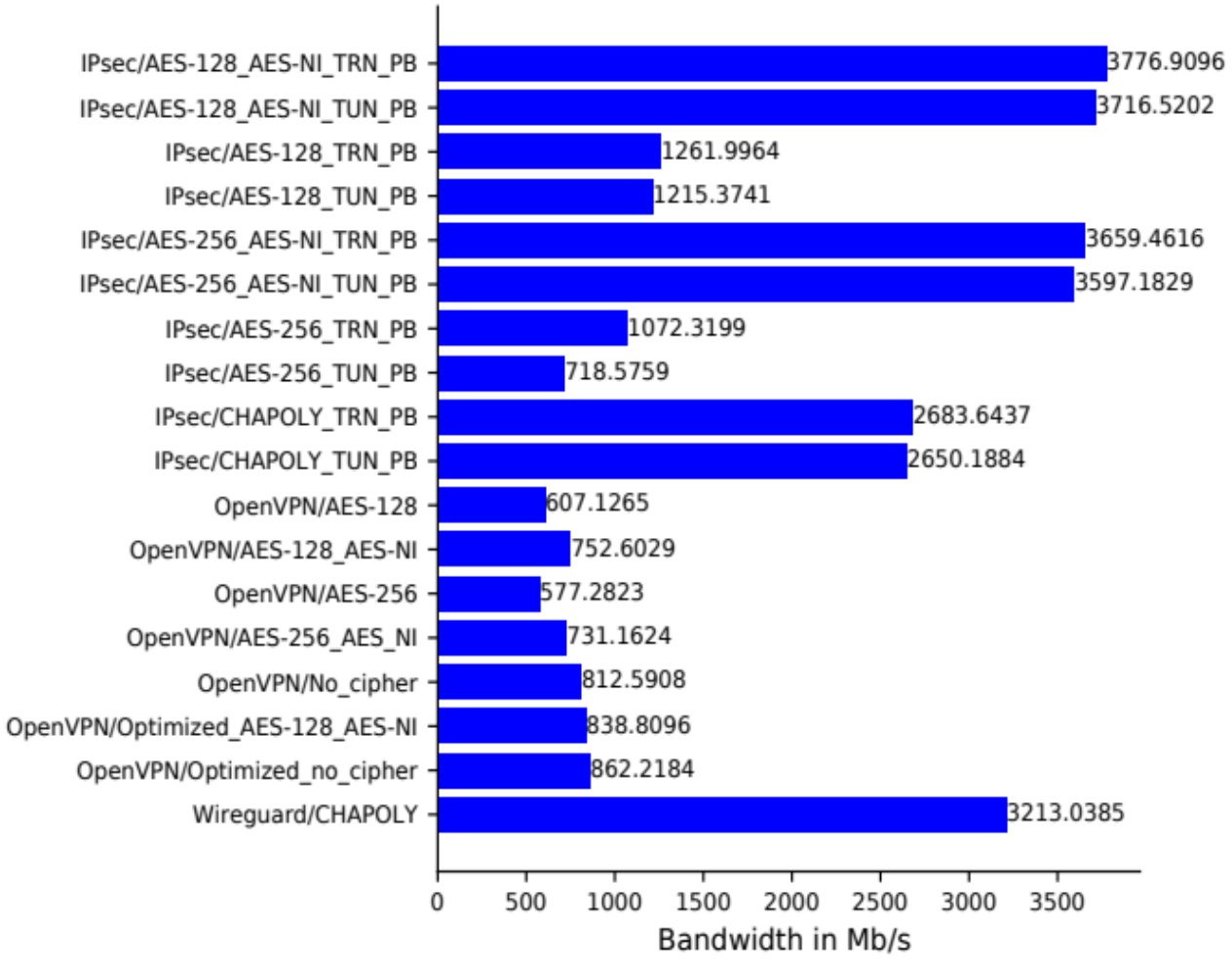


9 pav. WireGuard konfigūracija Linux aplinkoje [12]

Norint įvertinti WireGuard protokolo našumą lyginant su kitais virtualaus privataus tinklo protokolais, galime paanalizuoti šio protokolo interneto srauto pralaidumo palyginimą su kitais

protokolais. Kaip matyti iš 10 pav., WireGuard protokolas naudodamas ChaCha20-Poly1305 pasiekia reikšmingai didesnį pralaidumą lyginant su kitais protokolais, tokias kaip OpenVPN ar IPsec.

Pavyzdžiui, net ir optimizuotas OpenVPN retai viršija 860 Mb/s spartą. Tuo tarpu IPsec, naudojant AES šifravimą, nors ir yra greitesnis už OpenVPN, nusileidžia WireGuard kai nenaudojamas AES-NI aparatinis spartinimas. Šie rezultatai rodo, kad WireGuard protokolas yra tinkamesnis dinamiškoms bei realaus laiko reikalavimus atitinkančioms sistemoms, tokioms kaip autonominių transporto priemonių komunikaciniai tinklai.



10 pav. VPT tinklo protokolų palyginimas [13]

Nors WireGuard architektūra užtikrina paprastą ir efektyvų ryšio šifravimą, ji taip pat kelia tam tikrų saugumo iššūkių. Pagrindine priežastis yra viešųjų raktų atskleidimas, nes tai iš esmės gali leisti atsekti vartotojų IP adresus, jeigu nėra naudojami papildomi apsaugos mechanizmai. Taip pat, standartiškai protokolas nenumato vidinės autentifikacijos proceso, todėl tokiose sistemose kaip V2X būtina papildoma autentifikacija naudojant skaitmeninius sertifikatus ar naudojant patikimos platformos modulį (angl. *Trusted platform module*, TPM).

1.4.5. Aparatinės saugos modulis TPM

Šiame poskyryje nagrinėjamas TPM – aparatinės įrangos pagrindu veikiantis saugos sprendimas, skirtas kriptografinių raktų generavimui, saugojimui bei sistemų vientisumo patikrai.

TPM iš esmės yra izoliuota mikroschema, kurioje atliekamos pagrindinės kriptografinės operacijos. Ji turi nuosavą atmintį, procesorių bei atsitiktinių skaičių generatorių, todėl gali veikti nepriklausomai nuo pagrindinės sistemos procesoriaus ar operacinės aplinkos. Viena iš pagrindinių TPM funkcijų yra asimetrinių raktų porų generavimas, kai privatusis raktas sukuriamas tiesiogiai modulyje ir niekada neišvedamas į išorinę terpę, o viešasis raktas gali būti perduodamas tolesniam naudojimui, pavyzdžiui, registracijai ar autentifikacijai [14].

Kita svarbi funkcija yra duomenų pasirašymas bei šifravimas raktu, esančiu TPM viduje. Sistemos komponentas, norėdamas pasirašyti žinutę ar autentifikuotis, perduoda duomenis į TPM, kuris juos apdoroja ir grąžina rezultatą (pvz., skaitmeninį parašą), tačiau pats raktas niekada nebūna pasiekiamas iš išorės. TPM taip pat palaiko vientisumo patikros funkciją kai sistemos paleidimo metu gali būti skenuojami programiniai komponentai, o jų maišos reikšmės įrašomos į specialias saugumo registrų sritis (angl. *Platform Configuration Registers*). Šios reikšmės vėliau gali būti naudojamos patikrinimui, ar sistema buvo paleista autentiškoje, nepakeistoje būsenoje (angl. *trusted boot*).

TPM integruojamas į įterptinius įrenginius ar valdymo sistemas ir veikia kaip saugus vykdymo aplinkos (angl. *runtime*) komponentas, atskirtas nuo operacinės sistemos. TPM pagalba galima generuoti ir saugoti kriptografinius raktus, kurie niekada nepalieka modulyje esančios saugumo srities, taip apsaugant juos nuo programinių pažeidimų.

TPM ypač aktualus autonominių transporto priemonių tinkluose kaip aparatinės įrangos pagrindu veikiantis kriptografinio saugumo komponentas, galintis užtikrinti ilgalaikių asimetrinių raktų konfidencialumą. Tokiose sistemose, kuriose įrenginiai veikia atvirose ir potencialiai nepatikimose aplinkose, o saugus ir patikimas ryšys yra būtinas, aparatinio saugumo sprendimai tampa itin svarbūs. TPM leidžia ne tik saugiai laikyti raktus izoliuotoje atmintyje, bet ir vykdyti kriptografines operacijas modulyje jų neišduodant į išorę, taip užtikrinant dar aukštesnį apsaugos lygį. Pastaruoju metu vis plačiau pripažįstama TPM svarba diegiant saugumo priemones aparatiniu lygmeniu [15].

1.5. Analizės išvados

Atlikta analizė parodė, kad saugios komunikacijos užtikrinimui autonominių transporto priemonių tinkluose būtina taikyti pažangius ir našius kriptografinius sprendimus, kurie atitiktų realaus laiko ir sistemos efektyvumo reikalavimus. Standartizuotas ECDSA algoritmas leidžia užtikrinti patikimą transporto priemonių autentifikavimą bei perduodamų žinučių integralumą. Simetrinis ChaCha20-Poly1305 algoritmas, pasižymintis dideliu našumu net be aparatinio spartinimo, yra ypač tinkamas trumpiems ir dažniems duomenų perdavimo seansams, būdingiems V2X komunikacijai.

Taip pat analizuotas WireGuard protokolas išsiskiria modernia kriptografija, paprastu konfigūravimu ir dideliu našumu, kas leidžia jį taikyti dinamiškuose ir į realų laiką orientuotuose tinkluose. Be to, aparatinės įrangos pagrindu veikiantis TPM modulis suteikia galimybę užtikrinti ilgalaikių raktų saugumą bei aparatinio lygmens vientisumo tikrinimą, taip sustiprinant bendrą sistemos atsparumą kibernetinėms grėsmėms.

Apibendrinant galima daryti išvadą, kad analizuoti sprendimai yra tarpusavyje suderinami, o jų integravimas leidžia suformuoti saugų duomenų apsikeitimo kanalą, panaudojant WireGuard virtualaus tinklo protokolą. Toks sprendimas gali potencialiai padidinti duomenų perdavimo spartą, lyginant su tradiciniais V2X komunikacijos tinklais.

2. Autonominių transporto priemonių interneto srauto šifravimo metodas

Atlikus autonominių mašinų naudojamų tinklų analizę, nustatyta, kad transporto priemonių perduodamų duomenų vientisumo, konfidencialumo ir prieinamumo užtikrinimas yra itin svarbus. Šie duomenys naudojami navigacijos sistemoms, sąveikai su kelio infrastruktūros objektais bei ryšiui su kitomis transporto priemonėmis. Eliminavus žmogiškąjį faktorių iš transportavimo proceso, visa transporto priemonės kontrolė atliekama per interneto srautu perduodamas komandas. Todėl duomenų saugumas tampa kritiniu veiksniu užtikrinant autonominio transporto sistemos patikimumą ir atsparumą grėsmėms.

Norint apsaugoti per internetą perduodamą informaciją, būtina įdiegti patikimus kriptografinius sprendimus, paremtus stiprių šifravimo algoritmų taikymu. Vienas iš praktinių būdų šiam tikslui pasiekti – naudoti virtualaus privataus tinklo (VPT) technologijas, tokias kaip WireGuard ar OpenVPN, kurios leidžia formuoti saugų tunelį tarp tinklo mazgų. Be to, VPT taikymas sudaro sąlygas izoliuoti autonomines transporto priemones (pvz., prekių pristatymui skirtus robotus) nuo išorinių tinklų, kai reikalinga užtikrinti jų veikimą kontroliuojamoje, apsaugotoje aplinkoje [16].

Šiam metodui įgyvendinti pasirinkta naudoti WireGuard virtualaus privataus tinklo technologiją papildomai apsaugant autentifikacijos procesus naudojant TLS protokolą.

2.1. Siekiami rezultatai

2.1.1. Interneto srauto saugumo užtikrinimas

Autonominių transporto priemonių tinkluose saugumas yra esminis komponentas, lemiantis sistemos patikimumą bei atsparumą išorės grėsmėms. Kadangi perdavimo metu keičiamasi informacija, susijusia su transporto priemonės būsena, maršrutu ar valdymo komandomis, būtina užtikrinti trijų pagrindinių informacijos saugumo aspektų įgyvendinimą: konfidencialumą, vientisumą ir autentifikavimą.

Tam taikomi šiuolaikiniai šifravimo metodai: simetrinio srautinio šifravimo algoritmai (pvz., *ChaCha20-Poly1305*), asimetriniais skaitmeniniai parašo algoritmai (pvz., *ECDSA*) bei virtualaus privataus tinklo protokolai, tokie kaip WireGuard. Šie sprendimai leidžia apsaugoti duomenis nuo pasiklausymo, paketų modifikavimo ir žmogaus viduryje tipo atakų, suformuojant patikimą saugaus ryšio tunelį [17].

Be VPT protokolų, tam tikrose ryšio grandyse gali būti taikomas ir TLS protokolas, kuris užtikrina šifruotą ir autentifikuotą ryšį tarp taikomųjų programų lygmens komponentų. Tai ypač aktualu, kai autonominė transporto priemonė keičiasi informacija naudojant REST API sąsajas. TLS leidžia patikimai autentifikuoti tiek serverį, tiek klientą, pasitelkiant skaitmeninius sertifikatus, bei apsaugo duomenis nuo perėmimo, klastojimo ar pakeitimo.

Iš esmės TLS gali papildyti virtualus privataus tinklo tunelį, suteikdamas papildomą apsaugos sluoksnį aukštesniuose protokolo lygmenyse. Taip pat protokolas leidžia valdyti autentifikavimo bei šifravimo politiką tarp programinių komponentų, veikiančių toje pačioje fizinėje ar logiškai atskirtoje tinklo infrastruktūroje [18].

2.1.2. Taupus resursų naudojimas

Autonominiuose įrenginiuose dažnai naudojami mikrovaldikliai ar įterptiniai kompiuteriai su ribotais procesoriaus, atminties ir įvesties bei išvesties ištekliais. Todėl šifravimo metodai bei jų programinė realizacija turi būti kompaktiški, lengvi ir nereikalaujantys daug resursų. Siūlomas metodas remiasi šiuolaikiniais lengvo svorio protokolais, kurių algoritmai efektyviau išnaudoja ribotus sistemos resursus.

Tai reikštų, kad šifravimo operacijų efektyvumas turi būti vertinamas ne tik teoriniu, bet ir praktiniu požiūriu. Svarbu įvertinti kriptografinio kodo vykdymo laiką, atminties panaudojimą bei perdirbamų paketų kiekį. Tai leidžia užtikrinti, kad sistema neperkraus įrenginio ir galės veikti stabiliai realiu laiku.

2.1.3. Atsparumas tinklo trikdžiams ir ryšio netolygumui

Vienas iš metodo siekiamų tikslų yra užtikrinti patikimą duomenų perdavimą net ir tokiomis sąlygomis, kai tinklo prieinamumas yra ribotas arba kintantis. Autonominės transporto priemonės dažnai veikia aplinkose, kuriose ryšys palaikomas per mobiliojo ryšio ar belaidžio tinklo (WiFi) kanalus, pasižyminčius didele delsa ir ne visada pastovia kokybe. Dėl šios priežasties metodas turi būti pritaikytas veikti efektyviai esant sumažintam pralaidumui ir turėti minimalų papildomo tinklo srauto poreikį.

Taip pat siekiama, kad trumpalaikiai tinklo trikdžiai nesukeltų duomenų praradimo ar viso ryšio nutrūkimo. Todėl būtinas tinkamas srauto valdymas ir retransliacijos sprendimai, kurie būtų lengvi, nenaudotų perteklinių sisteminių išteklių ir tiktų realaus laiko sistemoms. Siūlomas metodas turi būti atsparus tinklo nestabilumui, pasitelkiant tokius sprendimus kaip atskiri, greitai sinchronizuojami šifravimo seansai, leidžiantys greitai atkurti ryšį be išorinių konfigūracijos veiksmų.

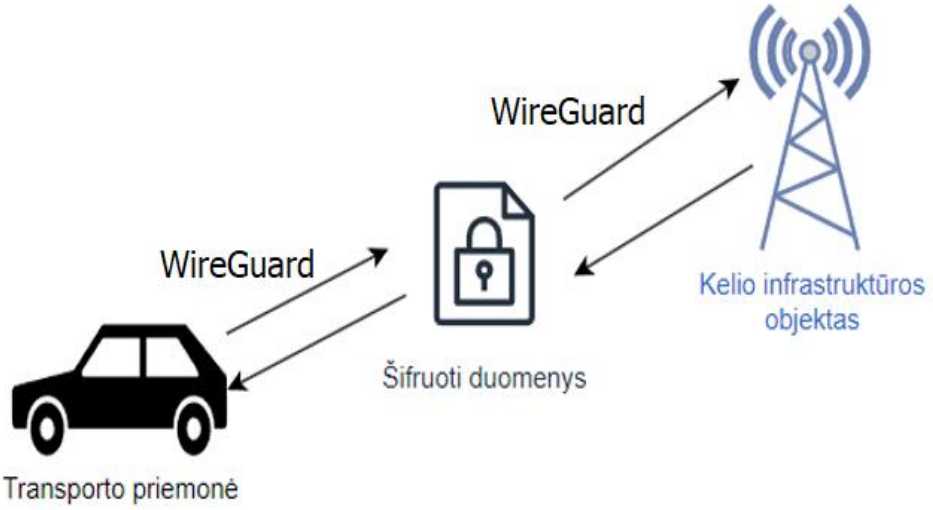
2.2. Siūloma architektūra

Siūloma architektūra numato saugaus ryšio užmezgimą tarp autonominės transporto priemonės ir kelio infrastruktūros objekto, pasitelkiant kelių lygių šifravimo bei autentifikacijos mechanizmus. Duomenų perdavimas tarp šių mazgų vykdomas per WireGuard tunelį, kuris užtikrina šifruotą, IP pagrindu veikiančią komunikacijos jungtį. Tačiau prieš užmezgant WireGuard tunelį, autentifikacijos duomenų apsikeitimo metu būtų naudojamas TLS šifravimas.

Ryšio autentifikacija būtų vykdoma naudojant statinių raktų modelį, kuriame kiekvienas ryšio dalyvis būtų identifikuojamas pagal viešojo rakto reikšmę. Siekiant padidinti raktų saugumą, siūloma naudoti TPM modulius, kurie leistų generuoti ir saugoti privačius raktus aparatinėje terpėje, taip sumažinant jų kompromitacijos riziką.

Papildomam programinės sąveikos saugumui užtikrinti siūloma naudoti TLS protokolą, kuris užtikrintų šifruotą komunikaciją tarp kelio infrastruktūros objekto ir autonominės transporto priemonės komponentų REST API sąsajoje. TLS autentifikavimas vyktų naudojant X.509 sertifikatus.

Žemiau vaizduojama koncepcinė architektūros schema, iliustruojanti šifruotos komunikacijos procesą tarp autonominės transporto priemonės ir kelio infrastruktūros objekto (žr. 11 pav.).



11 pav. Šifruotos komunikacijos modelis tarp transporto priemonės ir RSU

2.3. Metode naudojamos technologijos ir algoritmai

Šiame poskyryje aprašomos pagrindinės technologijos ir algoritmai, pasirinkti siūlomam saugaus duomenų perdavimo tarp autonominių transporto priemonių ir kelio infrastruktūros objekto sprendimui įgyvendinti. Kiekvienas komponentas buvo pasirinktas atsižvelgiant į reikalavimus saugumui, našumui, lengvam integravimui bei pritaikomumui ribotų resursų sistemose.

- **WireGuard** – modernus atvirojo kodo virtualaus privataus tinklo protokolas. Pasirinkta naudoti būtent šį protokolą dėl galimybės efektyviai šifruoti ryšį nenaudojant aparatinės spartinimo įrangos AES protokolui. Protokolas pagrįstas šiais algoritmais [12]:
  - **ChaCha20-Poly1305** – simetrinis šifravimas ir autentifikacija;
  - **Curve25519** – raktų mainų algoritmas;
  - **BLAKE2s** – greita maišos funkcija;
  - **HKDF** – raktų išvedimo funkcija.
- **TLS 1.3** – taikomas aplikacijų lygmens saugumui tarp autonominės transporto priemonės ir RSU komponentų. Jis naudojamas REST API komunikacijos apsaugai, per kurią vykdomi autentifikacijos ir registracijos procesai dar prieš VPT tunelio inicijavimą. Šis protokolas pasirinktas dėl kelių priežasčių [19]:
  - Plačiai palaikoma standartinė autentifikacijos sistema, paremta X.509 skaitmeniniais sertifikatais;
  - Patikima šifruoto kanalo užmezgimo galimybė net nesaugioje tinklo aplinkoje;
  - Lengva integracija su esamomis technologijomis, tokiomis kaip FastAPI.

- **TPM** – tai aparatinės įrangos pagrindu veikiantis saugumo komponentas, skirtas asimetrinių raktų generavimui ir saugojimui. TPM pasirinktas dėl gebėjimo užtikrinti, kad privatūs raktai niekada nepalieka saugios modulio terpės, net ir esant galimai nesaugiai operacinei aplinkai. Ši funkcija ypač svarbi VPT tunelio privatumo užtikrinimui. Naudojant TPM [20]:
    - Privatūs raktai generuojami tiesiogiai modulyje ir nėra eksportuojami;
    - Galima vykdyti kriptografines operacijas pačiame modulyje;
    - Užtikrinamas įrenginio tapatumo patikimumas bei atitiktis nulinio pasitikėjimo architektūros principams.
- **FastAPI** – pasirinkta kaip pagrindinė REST API kūrimo platforma dėl savo greito veikimo, asinchroninio darbo galimybių ir natūralios integracijos su TLS. Ji naudojama API serverio įgyvendinimui RSU pusėje, per kurį transporto priemonės registruojasi ir apsikeičia reikalinga konfigūracija. FastAPI privalumai [21]:
    - Lengvas integravimas su *uvicorn* serveriu, *OpenSSL* biblioteka, TLS skaitmeniniais sertifikatais;
    - Palaiko greitą, asinchroninę komunikaciją tarp komponentų.
- **Papildomai panaudotos programinės priemonės bei bibliotekos:**
    - *wg, wg-quick* WireGuard sąsajų valdymui ir tunelio inicijavimui;
    - *OpenSSL* TLS skaitmeninių sertifikatų generavimui ir testavimui;
    - *tpm2-tools* TPM modulių raktų generavimo, skaitymo ir saugojimo komandoms vykdyti;
    - *uvicorn, FastAPI, requests* REST API veikimui, serverio paleidimui ir duomenų perdavimui.

2.4. Sistemos komponentų sąveika

Šiame poskyryje aprašoma siūlomo metodo veikimo seka, atskleidžianti, kaip autonominė transporto priemonė užmezga saugų ryšį su kelio infrastruktūros objektu, pasitelkiant WireGuard virtualaus privataus tinklo tunelį, TLS protokolą bei TPM modulį.

Sąveika tarp komponentų siūloma išskaidyti į šiuos etapus:

1. **Rakto generavimas autonominėje transporto priemonėje.** Transporto priemonė inicijuoja WireGuard ryšio paruošimą, pateikdama užklausą savo TPM moduliui sugeneruoti privatų raktą. TPM viduje raktas sugeneruojamas ir saugomas saugioje, nuo OS izoliuotoje atmintyje. Iš TPM išvedamas tik atitinkantis viešasis raktas, kuris vėliau perduodamas kelio infrastruktūros objektui RSU.
2. **TP registracija kelio infrastruktūros objekte RSU.** Naudodama TLS protokolu apsaugotą kanalą, transporto priemonė perduoda RSU serveriui savo viešąjį raktą bei skaitmeninį sertifikatą naudojant API sąsają. Šis perdavimas vyksta prieš virtualaus privataus tinklo tunelio sukūrimą, siekiant atlikti autentifikaciją ir paruošti reikiamą konfigūraciją virtualaus privataus tinklo užmezgimui.
3. **Sertifikato patikrinimas.** Gavęs transporto priemonės skaitmeninį sertifikatą, kelio infrastruktūros objektas (RSU) atlieka jo autentiškumo ir galiojimo patikrą, naudodamas lokaliai

saugomą sertifikavimo centro (CA) šakninį sertifikatą (angl. *root certificate)*. Sėkmingai patvirtinus transporto priemonės tapatybę, RSU vykdo šiuos veiksmus:

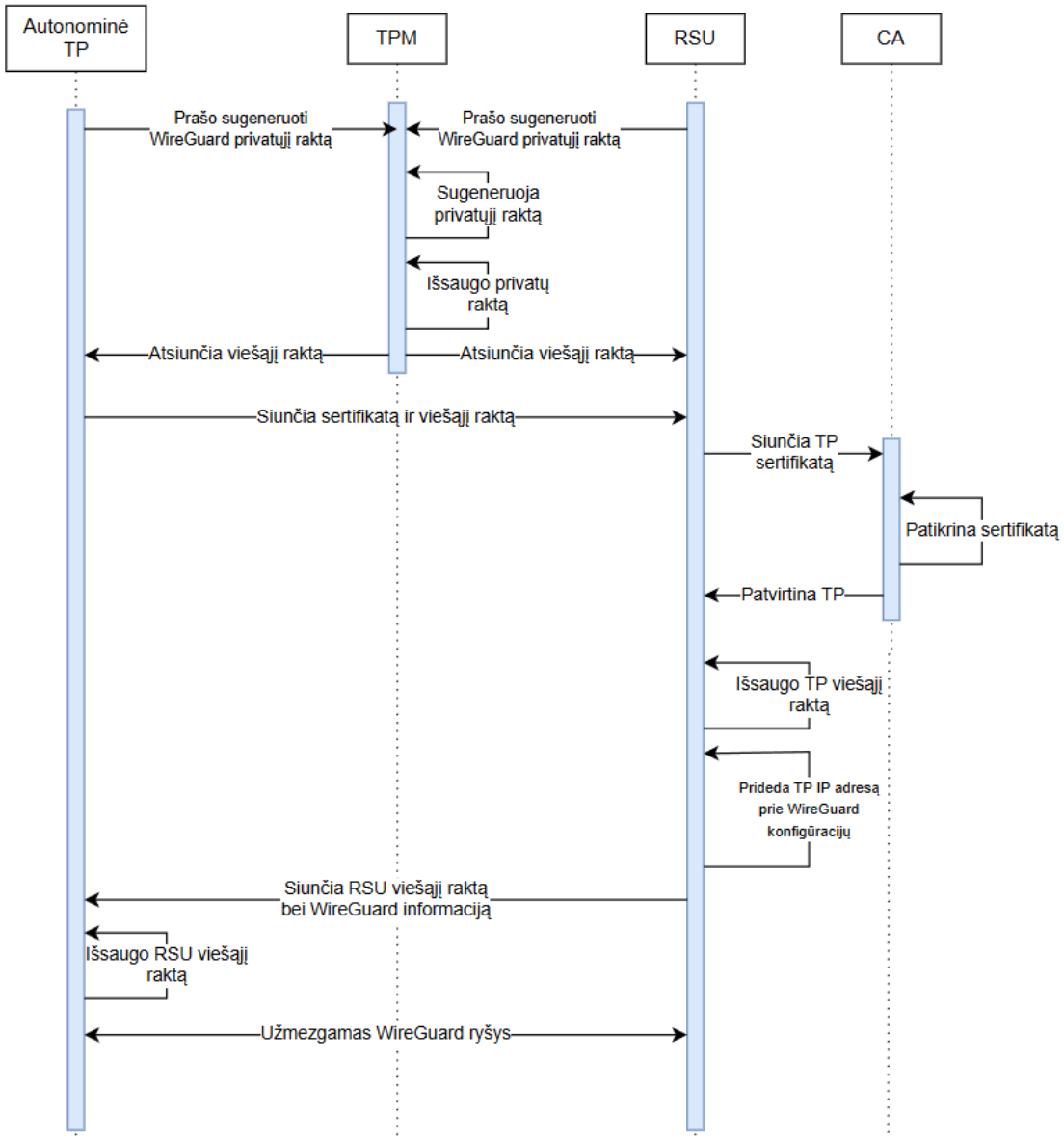
3.1. Įrašo transporto priemonės viešąjį raktą į WireGuard konfigūraciją.

3.2. Priskiria laisvą IP adresą autonominei transporto priemonei.

3.3. Išsaugo transporto priemonės identifikaciją sistemoje.

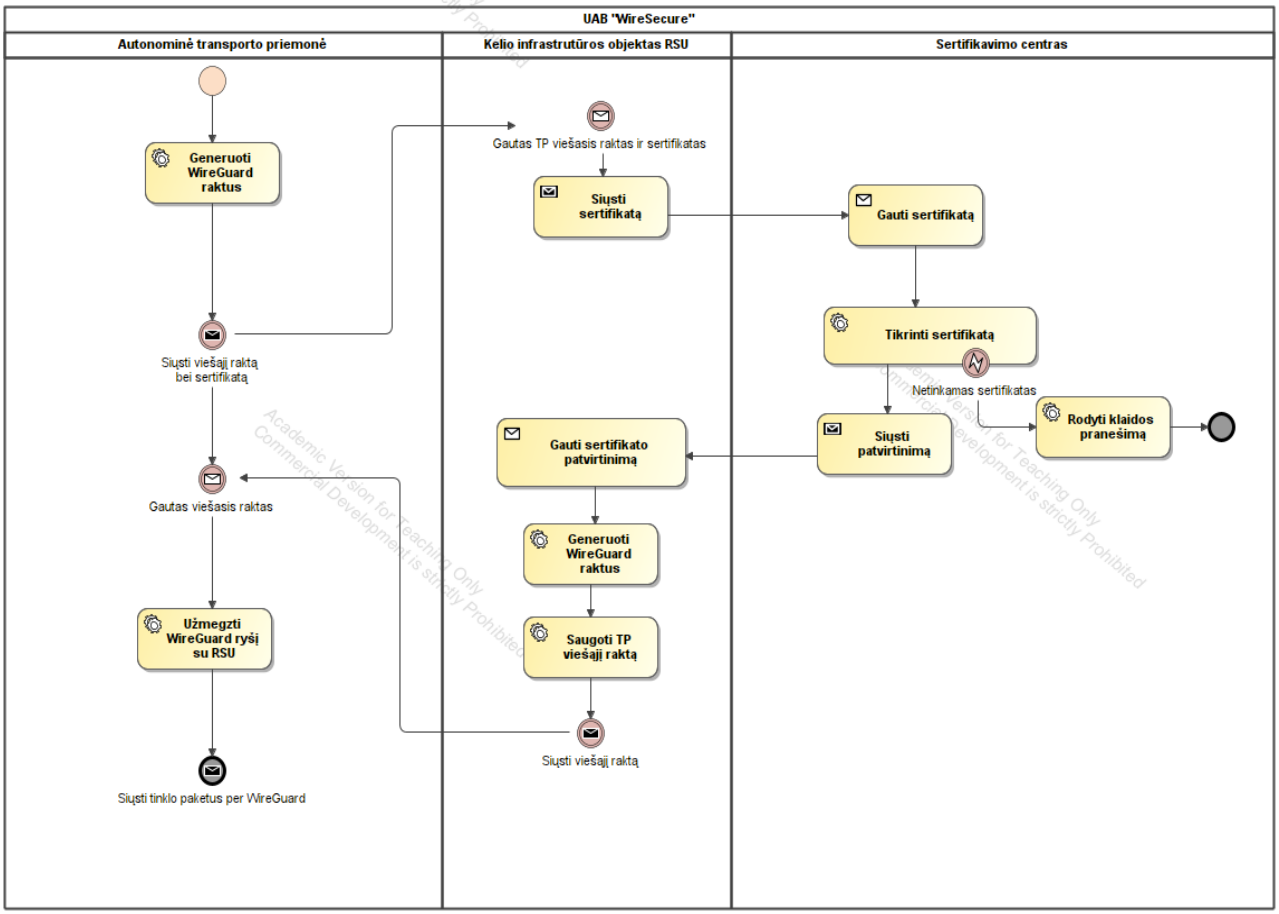
4. **Atsakymas iš kelio infrastruktūros objekto RSU ir WireGuard tunelio inicijavimas**. RSU grąžina transporto priemonei savo viešąjį raktą ir būtiną informaciją, pavyzdžiui, laisvą IP adresą. Transporto priemonė išsaugo gautus duomenis ir naudodama savo TPM saugomą privatų raktą, užmezga saugų WireGuard VPT tunelį.
5. **Ryšio užmezgimas.** Įgalinus virtualaus privataus tinklo tunelį, galima pradėti siųsti šifruotus duomenis saugiu kanalu. Šis tunelis naudojamas informacijos apsikeitimui tarp kelio infrastruktūros objekto RSU ir autonominės transporto priemonės.

Šio metodo veiksmų seką pavaizduota žemiau (žr. 12 pav.).



12 pav. Siūlomas autentifikacijos mechanizmas WireGuard ryšiui užmegzti

Taip pat pateikiama veiklos diagrama (žr. 13 pav.), kuri iliustruoja metode siūlomą autentifikacijos mechanizmą. Joje vaizduojami pagrindiniai veiklos žingsniai, įtraukiantys raktų generavimą, skaitmeninio sertifikato patikrinimą bei saugaus WireGuard ryšio užmezgimą tarp autonominės transporto priemonės ir kelio infrastruktūros objekto. Diagrama aiškiai išskiria atsakomybes tarp sistemos komponentų ir atspindi loginę autentifikacijos proceso eigą.



13 pav. Autonominės transporto priemonės WireGuard kanalo procesas

Svarbu pažymėti, kad autonominė transporto priemonė privalo turėti galiojantį skaitmeninį sertifikatą dar prieš užmezgant saugų WireGuard ryšį. Šis sertifikatas gali būti išduodamas transporto priemonės gamybos metu arba įdiegtas pirmojo sistemos paleidimo metu. Siekiant didesnio lankstumo ir centralizuoto valdymo, gali būti taikomas pažangesnis metodas – sertifikato išdavimas registracijos metu, naudojant viešųjų raktų infrastruktūrą (PKI).

Tai pat akcentuojama tai, kad visa autentifikacija turi būti vykdoma naudojant saugų TLS protokolą siekiant apsisaugoti nuo galimų žmogaus viduryje atakų ir užtikrinti perduodamų duomenų konfidencialumą bei vientisumą.

2.5. Metodo išvados

Apibendrinant galima teigti, kad siūlomas saugaus ryšio metodas, paremtas WireGuard virtualaus privataus tinklo tuneliu, TLS autentifikacijos mechanizmu bei TPM aparatinio saugumo moduliu, sudaro patikimą pagrindą autonominių transporto priemonių duomenų apsaugai. Parinkti sprendimai pasižymi našumu, saugumu ir pritaikomumu ribotų resursų sistemose.

Sistemos architektūra išskaidyta į loginius komponentus, kurių sąveika leidžia užtikrinti šifruotą, autentifikuotą ir atsparesnį trikdžiams ryšį. Ši metodika suformuoja teorinį pagrindą tolimesniam eksperimentiniam vertinimui, kuriame bus tiriamas ryšio našumas bei apsaugos efektyvumas.

3. Autonominių transporto priemonių interneto srauto šifravimo metodo prototipas

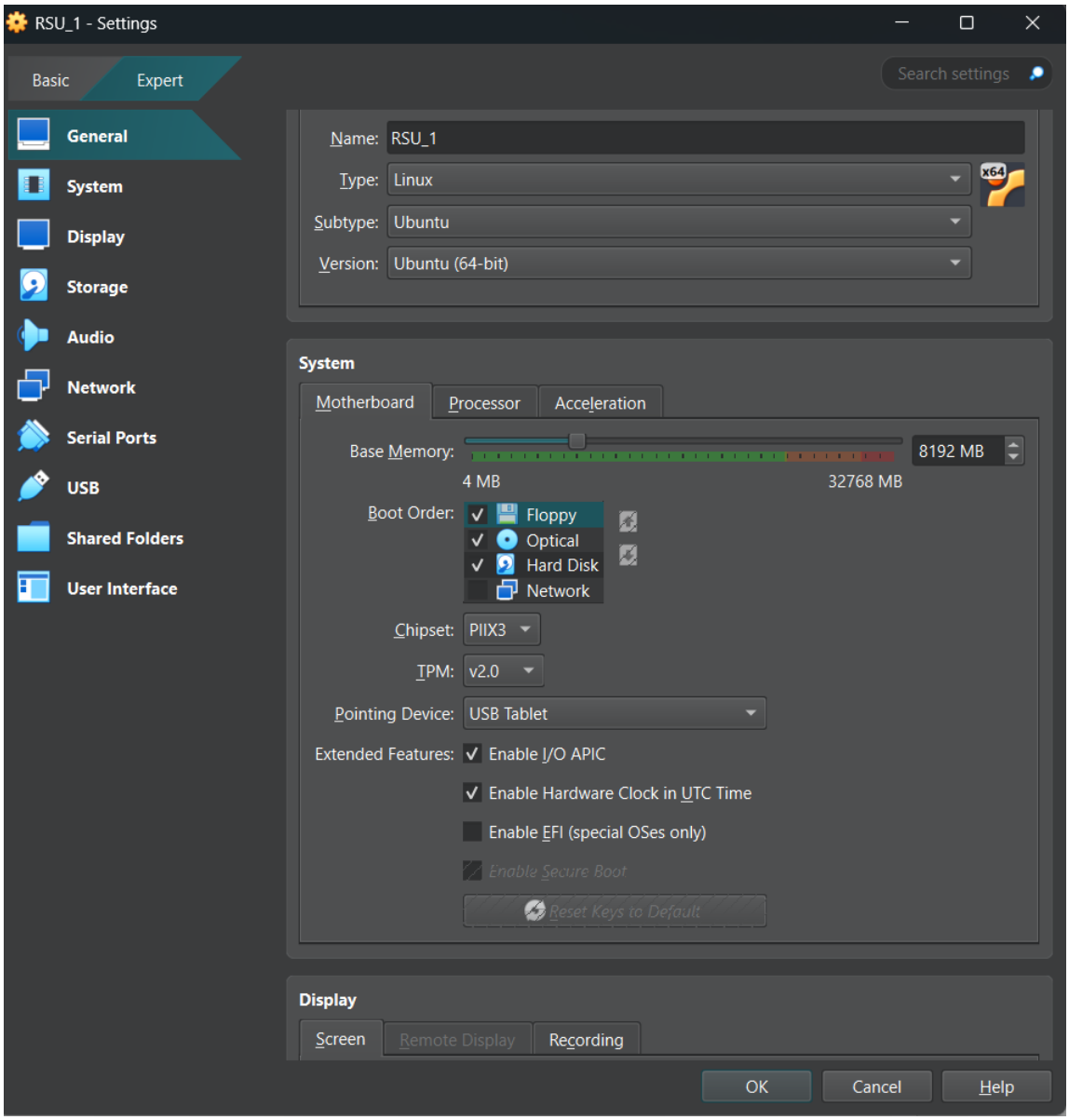
Šiame skyriuje aprašomas suprojektuotas autonominės transporto priemonės ir kelio infrastruktūros objekto ryšio prototipo modelis. Juo siekiama pademonstruoti, kaip šifravimo bei autentifikacijos sprendimai – tokie kaip WireGuard, TLS protokolas ir TPM modulis gali būti integruoti siekiant užtikrinti konfidencialų, patikimą ir atsparų ryšio kanalą tarp autonominės transporto priemonės ir RSU.

3.1. Prototipo architektūra

Prototipas buvo įgyvendintas naudojant dvi pagrindines virtualiąsias mašinas: CAR_1 (autonominę transporto priemonę) ir RSU_1 (kelio infrastruktūros objektą). Virtualiosioms mašinoms valdyti pasirinktas Oracle VirtualBox hipervizorius, kuriame įdiegta Linux Lubuntu operacinė sistema. Sukurtas modelis pademonstruoja, kaip CAR_1 saugiai prisijungia prie RSU_1, naudodama TLS protokolu apsaugotą API užklausą. Per šią užklausą perduodami pagrindiniai parametrai, reikalingi WireGuard tinklo konfigūracijai, leidžiantys vėliau užmegzti šifruotą virtualaus privataus tinklo tunelį tarp abiejų mazgų.

3.1.1. Virtualiosios mašinos

Kelio infrastruktūros objektą imituojanti virtualioji mašina sukonfigūruota kaip pagrindinis tinklo mazgas, atsakingas už saugią sąveiką su autonomine transporto priemone. Šiai virtualiai mašinai prototipo modelyje yra priskirti 4 procesoriaus branduoliai bei 8GB darbinės atminties. Ši konfigūracija leidžia vienu metu apdoroti kelias autonominės transporto priemonės užklausas, įskaitant skaitmeninio sertifikato tikrinimą, WireGuard raktų valdymą bei saugaus tunelio suformavimą. Šios virtualios mašinos konfigūracinis langas pateikiamas žemiau (žr. 14 pav.).



14 pav. RSU_1 virtualios mašinos konfigūraciniai parametrai

RSU_1 virtualioje mašinoje veikia serverio programa, sukurta naudojant Python programavimo kalbą ir pagrįsta „FastAPI“ karkasu. Ši programa atlieka keletą esminių funkcijų:

1. **Sertifikavimų validacija.** Sertifikato validacija. RSU_1 įrenginys priima autonominės transporto priemonės pateiktą skaitmeninį sertifikatą bei WireGuard viešąjį raktą. Sertifikato autentiškumas tikrinamas lokaliai, naudojant į RSU_1 virtualiąją mašiną įdiegtą sertifikavimo centro (CA) šakninį sertifikatą. Tikrinimo procesas įgyvendintas pasitelkiant *OpenSSL* biblioteką, leidžiančią patikrinti parašo autentiškumą bei sertifikato galiojimą.
2. **Viešųjų raktų valdymas.** Sėkmingai patikrinus sertifikatą, RSU_1 išsaugo transporto priemonės viešąjį raktą savo vidinėje duomenų bazėje. Šis raktas vėliau naudojamas WireGuard ryšio konfigūracijai. Kadangi raktas jau susietas su konkrečia transporto

priemone, jo nereikia pakartotinai pateikti, o tai sumažina autentifikacijos proceso sudėtingumą ateityje.

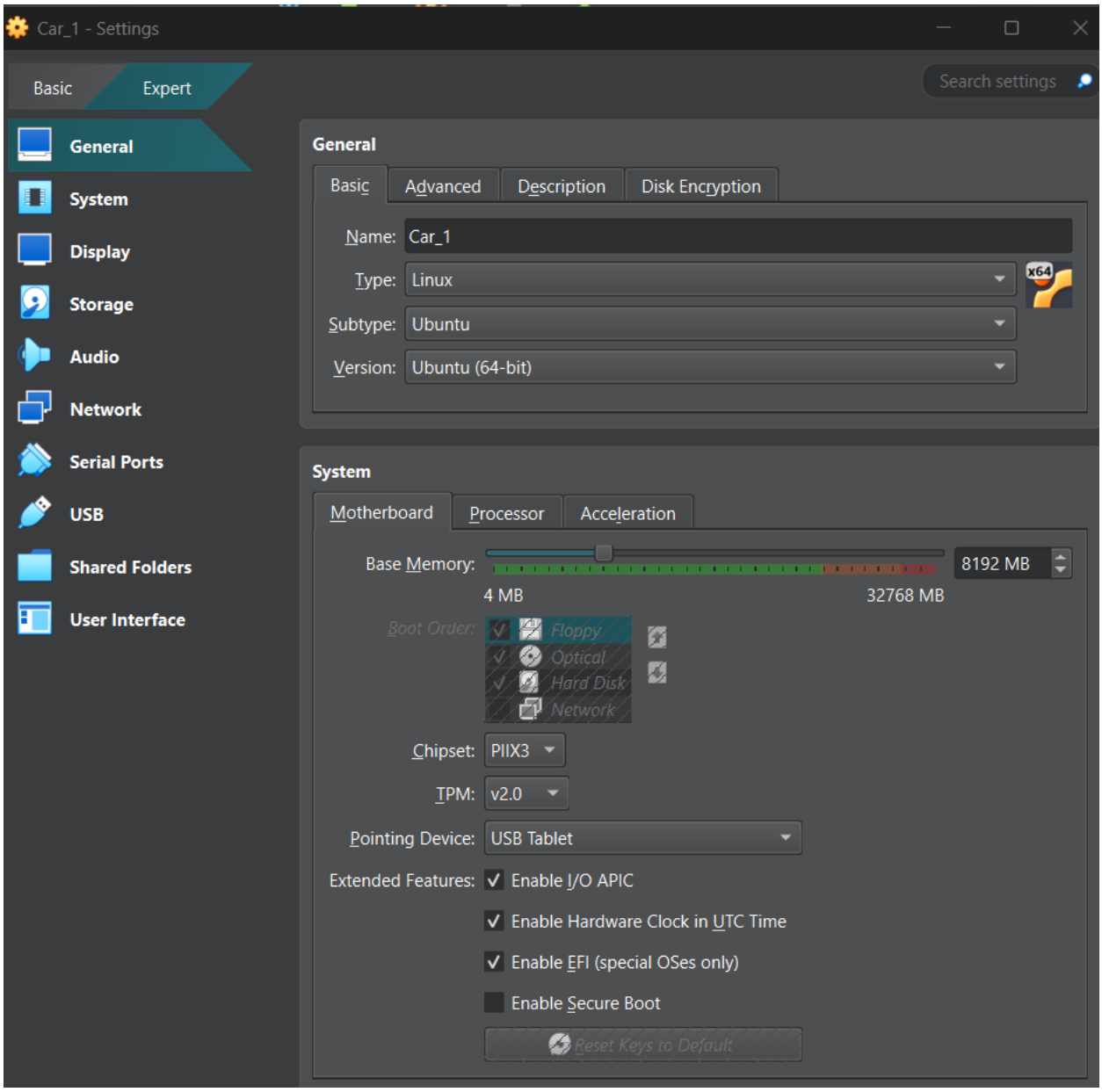
3. **WireGuard tinklo konfigūravimas:** RSU_1 įrenginyje sugeneruojami būtini WireGuard parametrai – leidžiamų IP adresų intervalai, tinklo sąsajos identifikatoriai bei asimetrinių raktų pora. Privačiojo rakto generavimas atliekamas naudojant TPM modulį, kuris užtikrina, kad raktas niekuomet nepaliktų saugios aparatinės aplinkos ir nebūtų saugomas atviro tekstinio formato pavidalu. WireGuard viešasis raktas, gautas iš TPM modulyje esančio privataus rakto, kartu su reikalingais konfigūracijos parametrais perduodamas transporto priemonei, siekiant įgalinti VPT jos pusėje.

Šiame prototipe ypatingas dėmesys skiriamas saugumo užtikrinimui. Naudojami sertifikavimo centro (CA) pasirašyti skaitmeniniai sertifikatai leidžia patikimai identifikuoti įrenginius ir užtikrinti, kad prie tinklo galėtų prisijungti tik autorizuoti tinklo mazgai. Tuo tarpu WireGuard naudojami raktai užtikrina duomenų konfidencialumą ir vientisumą. Jei transporto priemonė pateikia netinkamą arba negaliojantį sertifikatą, RSU_1 grąžina klaidos pranešimą, atmeta autentifikacijos bandymą ir neteikia reikiamos prisijungimo prie WireGuard tinklo informacijos.

Be to, RSU_1 geba aptarnauti kelis klientus vienu metu. Serverio programa gali apdoroti daug registracijos užklausų be reikšmingo našumo kritimo, todėl yra tinkama naudoti realiomis sąlygomis, kai prie vieno RSU jungiasi kelios autonominės transporto priemonės.

Svarbu pažymėti, kad RSU virtualiosios mašinos gali būti lengvai klonuojamos. Visos jos gali naudoti bendrą šakninį sertifikatą, tačiau turi skirtingus WireGuard konfigūracijos parametrus – skiriasi IP adresai ir raktų poros.

Antroji virtuali mašina, pavadinta CAR_1, sukurta taip, kad realizuotų pačios autonominės transporto priemonės funkcionalumą bei komunikaciją su kelio infrastruktūros objektu naudojant WireGuard. Kaip ir RSU_1, ši mašina įdiegta naudojant Oracle VirtualBox hipervizorių bei veikia Linux Lubuntu operacinėje sistemoje. CAR_1 virtualiajai mašinai priskirti 4 procesoriaus branduoliai ir 8 GB darbinės atminties. Tai leidžia efektyviai vykdyti autentifikacijos bei šifruoto duomenų perdavimo procesus be didelės sistemos apkrovos. Mašinos pagrindiniai konfigūracijos parametrai pateikiami žemiau (žr. 15 pav.).



15 pav. CAR_1 virtualios mašinos konfigūraciniai parametrai

CAR_1 sistemoje realizuotos dvi pagrindinės funkcijos:

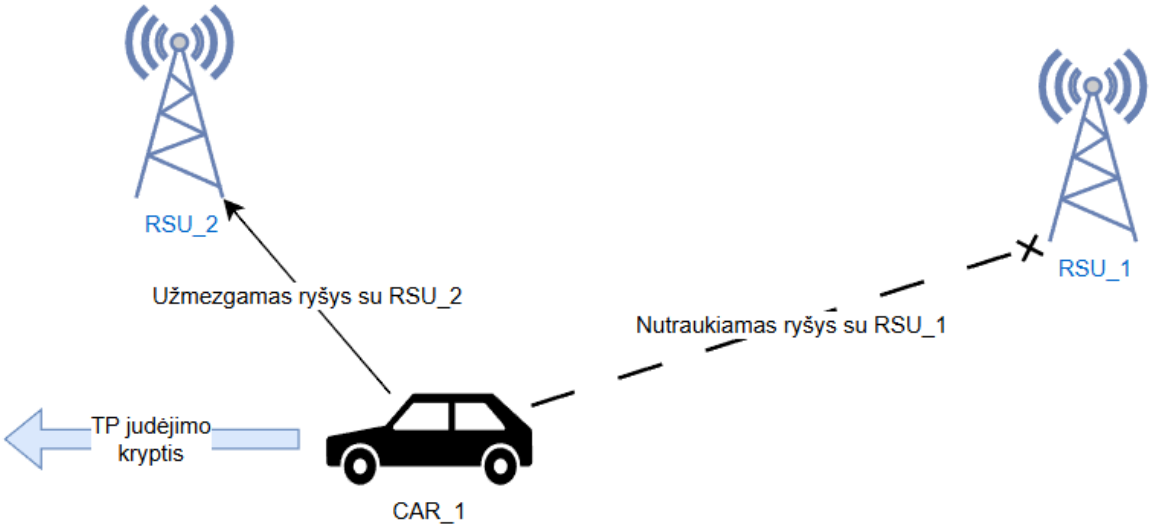
- autentifikacija su RSU;
- šifruotų duomenų perdavimas per užmegztą WireGuard tunelį.

Pagrindinėms virtualiosios mašinos CAR_1 funkcijoms realizuoti naudojamos dvi atskiros kliento programos: viena atsakinga už registraciją ir autentifikaciją per TLS apsaugotas REST API užklausas, kita – už duomenų siuntimą tik per WireGuard tinklą. Ši architektūra padeda atskirti saugumo funkcijas ir išlaikyti vidinę logiką atskirą nuo išorinio pasaulio, nes duomenų siuntimo programa komunikuoja tik per saugų VPT tunelį.

Vienas iš esminių komponentų yra *TPM* modulis, integruotas į RSU_1 virtualiąją mašiną. Jis naudojamas saugiam privačių raktų generavimui ir saugojimui. Šie raktai niekada nėra tiesiogiai

prieinami operacinei sistemai ir gali būti naudojami tik pasitelkiant TPM operacijas, o tai ženkliai padidina atsparumą kibernetinėms grėsmėms. Vadovaujantis WireGuard veikimo principais, viešasis raktas yra deterministiškai išvedamas iš privataus, ir būtent jis yra perduodamas autentifikacijos metu. Iš esmės vienintelis būdas pasiekti privatųjį raktą yra kompromituoti patį *TPM* modulį, o tam reikalinga fizinė prieiga prie įrenginio arba pažangios atakos, nukreiptos prieš aparatinės apsaugos mechanizmus.

CAR_1 taip pat įgyvendina dinaminio RSU keitimo funkciją. Jei dabartinis RSU tampa neprieinamas, sistema automatiškai inicijuoja perjungimą į kitą prieinamą infrastruktūros objektą. Tai leidžia palaikyti nuolatinį ir patikimą saugų tunelį net esant tinklo pokyčiams – o tai ypač aktualu mobiliose autonominėse sistemose. Šio scenarijaus koncepcinis modelis pateikiamas žemiau (žr. 16 pav.).



16 pav. Dinaminis transporto priemonės persijungimas tarp RSU objektų.

3.1.2. Elektroniniai sertifikatai

Sugeneruoti skaitmeniniai sertifikatai naudojami kaip pagrindinis autentifikacijos mechanizmas tarp autonominės transporto priemonės (CAR_1) ir kelio infrastruktūros objekto (RSU_1). Visa registracijos procedūra vyksta per TLS protokolu apsaugotą REST API sąsają. Transporto priemonė, inicijuodama registraciją, kartu su savo viešuoju raktu pateikia skaitmeninį sertifikatą, kuris RSU pusėje patikrinamas naudojant sertifikavimo centro (CA) šakninį sertifikatą.

Šis autentifikacijos modelis leidžia užtikrinti, kad prie RSU_1 WireGuard tinklo galėtų prisijungti tik tie įrenginiai, kurių sertifikatai atitinka nustatytą pasitikėjimo grandinę. Jei pateiktas sertifikatas yra galiojantis ir pasirašytas patikimo sertifikavimo centro (CA), RSU_1 suteikia leidimą tęsti WireGuard tunelio konfigūravimo procesą. Priešingu atveju autentifikacija laikoma nesėkminga, o prisijungimo procedūra nutraukiama.

Demonstraciniame prototipe, realizuotame virtualiųjų mašinų aplinkoje, sertifikatai naudojami šiuose autentifikacijos etapuose:

1. Autonominė transporto priemonė (CAR_1) perduoda savo skaitmeninį sertifikatą kartu su WireGuard viešuoju raktu kelio infrastruktūros objektui (RSU_1).
2. RSU_1, naudodama *OpenSSL* biblioteką ir lokaliai saugomą sertifikavimo centro (CA) šakninį sertifikatą, atlieka sertifikato autentiškumo ir jo galiojimo patikrą.
3. Patvirtinus sertifikato galiojimą, atsiųstas viešasis raktas susiejamas su konkrečia transporto priemone ir įtraukiamas į WireGuard konfigūraciją.
4. Vėlesnių sesijų metu tas pats sertifikatas gali būti naudojamas autentifikacijai, jei dar nėra pasibaigęs jo galiojimo terminas.

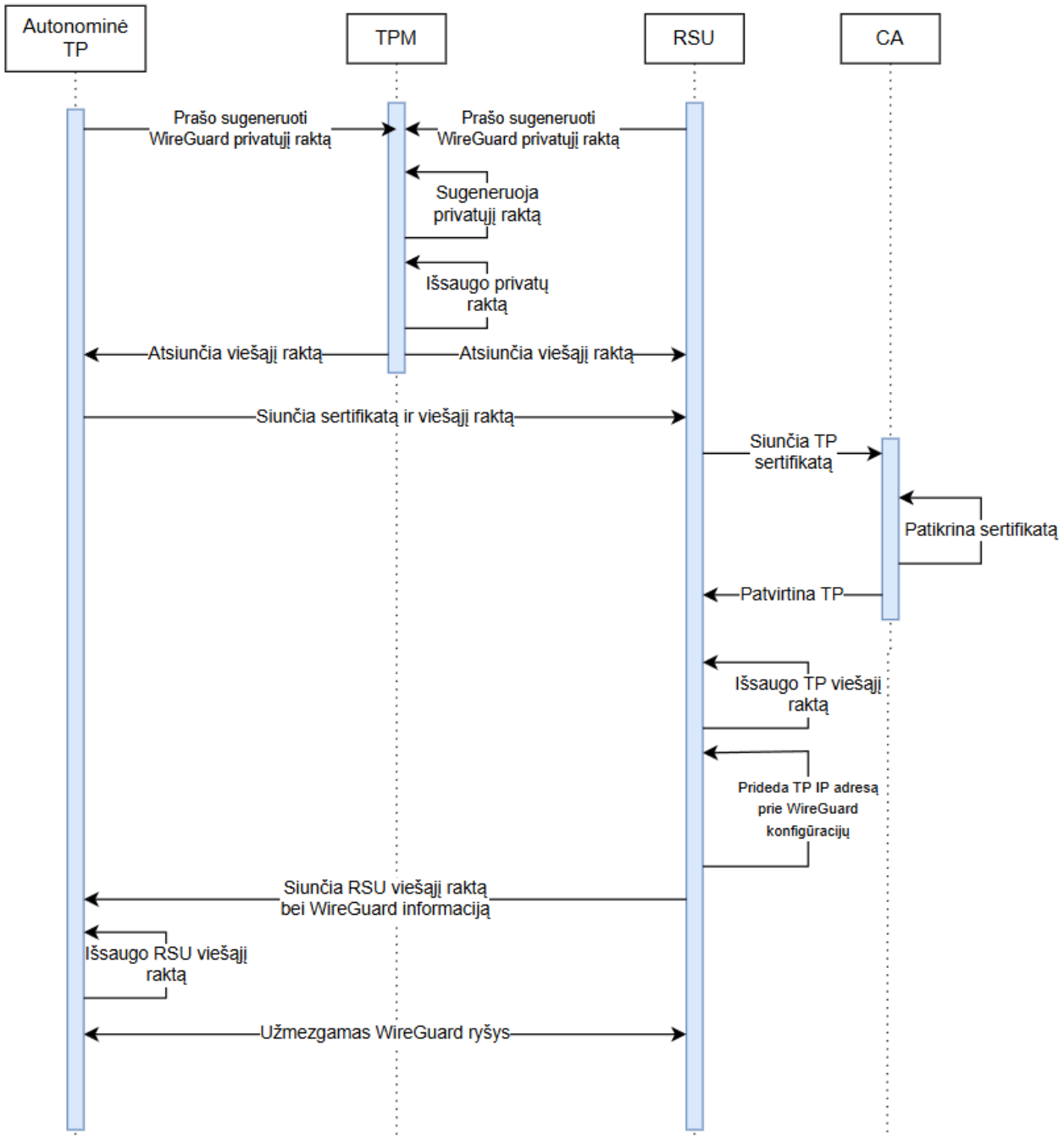
Toks sertifikatais grįstas autentifikacijos modelis yra lankstus ir gali būti lengvai išplėstas realiose sistemose, pavyzdžiui, integruojant automatizuotą sertifikatų galiojimo stebėseną ar atnaujinimą.

3.1.3. TPM raktų generavimas ir naudojimas

TPM modulis čia atlieka esminį vaidmenį užtikrinant raktų saugumą autonominėje transporto priemonėje. Prototipe TPM naudojamas tam, kad sugeneruotas WireGuard privatus raktas niekada nebūtų laikomas atviru tekstu ar prieinamas failų sistemoje. Vietoj to, raktas sugeneruojamas vieną kartą, vėliau užšifruojamas ir saugomas kaip .ctx arba .enc failas.

Kai sistemai reikia suformuoti WireGuard tunelį, užšifruotas raktas iššifruojamas tik programos veikimo metu taikant TPM teikiamas operacijas. Raktas naudojamas tik darbinėje atmintyje ir nėra išsaugomas diske atviro teksto pavidalu. Tai leidžia užtikrinti, kad net esant sistemos kompromitavimui, tikrasis privatusis raktas liktų apsaugotas. Šiuo atveju TPM modulis saugo ne tik rakto kilmę, bet ir prieigos valdymą. Tokiu būdu pasiekiamas balansas tarp saugumo lygio ir praktinio prototipo įgyvendinimo, išlaikant galimybę programiškai inicijuoti WireGuard ryšį realiuoju laiku.

Šiame skyriuje, kaip praktinio prototipo realizacijos pagrindas, pateikiama pakartotinai rodoma sekos diagrama (žr. 17 pav.), iliustruojanti saugaus WireGuard ryšio užmezgimą tarp autonominės transporto priemonės ir RSU.



17 pav. Autonominės transporto priemonės autentifikacijos ir WireGuard ryšio užmezgimo seka

3.2. Prototipo išvados

Sukurtas prototipas parodė, kad saugi komunikacija tarp autonominės transporto priemonės (CAR_1) ir kelio infrastruktūros objekto (RSU_1) gali būti efektyviai įgyvendinta naudojant dviejų lygių saugumo modelį, kurio metu registracijos procesas būtų apsaugomas TLS protokolu, o duomenų perdavimo kanalas – WireGuard. CAR_1 virtualiojoje mašinoje realizuotos dvi atskiros kliento programos leidžia aiškiai atskirti autentifikacijos funkciją nuo šifruoto duomenų perdavimo, o integruotas TPM modulis užtikrina saugų privačių raktų generavimą bei naudojimą. Šie raktai nėra prieinami operacinei sistemai ar saugomi diske nešifruotu pavidalu.

RSU_1 virtualiojoje mašinoje, pasitelkiant lokaliai saugomą sertifikavimo centro (CA) šakninį sertifikatą, atliekamas pateikto sertifikato autentiškumo ir galiojimo tikrinimas taip užtikrinant, kad tik autorizuotos transporto priemonės gali prisijungti prie tinklo. Sukurtas prototipas taip pat palaiko dinaminį RSU objektų keitimą, pavyzdžiui, CAR_1 gali automatiškai prisijungti prie alternatyvaus RSU, jei pirminis tampa nepasiekiamas. Toks sprendimas leidžia užtikrinti nepertraukiamą ir patikimą ryšį, kuris yra būtinas mobiliose sistemose.

Taip pat ši prototipo architektūra yra lengvai plečiama: tiek kelio infrastruktūros objektų, tiek autonominių transporto priemonių virtualiosios mašinos gali būti klonuojamos, o WireGuard autentifikacijos proceso konfigūracijos pritaikomos individualiai, naudojant bendrą iš anksto sugeneruotų sertifikatų rinkinį. Nors šis prototipas remiasi statiniu sertifikatų paskirstymu, realiose sistemose būtų taikoma išplėstinė viešojo rakto infrastruktūra (PKI), kurioje sertifikatai būtų išduodami ir tikrinami pasitelkiant sertifikavimo centrą (CA). Tokiu būdu būtų užtikrinamas dinamiškas raktų valdymas, sertifikatų galiojimo kontrolė ir centralizuotas pasitikėjimo modelis. Toks sprendimas yra tinkamas tiek eksperimentiniam testavimui, tiek praktiniam diegimui realiose mobilumo platformose, kuriose svarbus saugus, patikimas ir nuolat palaikomas ryšys tarp transporto priemonių ir infrastruktūros komponentų.

4. Autonominių transporto priemonių interneto srauto šifravimo metodo eksperimentinis tyrimas

Šiame skyriuje pateikiami eksperimentiniai rezultatai, pagrindžiantys sukurto saugaus interneto srauto šifravimo metodo veikimą ir tinkamumą autonominių transporto priemonių komunikacijai su kelio infrastruktūros objektais. Tyrimas atliktas naudojant sukurtą prototipo modelį, kuriame integruotas WireGuard virtualaus privataus tinklo tunelis, TLS protokolu apsaugota autentifikacija bei TPM modulis, skirtas raktų generavimui ir apsaugai.

Eksperimento vertinimo metu analizuojami komponentų veikimo principai, stebima reali tinklo sąveika tarp transporto priemonės ir kelio infrastruktūros objekto bei vertinami interneto pralaidumo rodikliai.

Tyrimo tikslas yra įvertinti, ar siūlomas sprendimas atitinka realaus laiko, saugumo ir efektyvumo reikalavimus, būdingus šiuolaikinėms mobilumo sistemoms.

4.1. Kelio infrastruktūros objekto (RSU) serveris

Kelio infrastruktūros objektas realizuotas kaip virtuali mašina, kurioje įdiegta Python programavimo kalba sukurta autentifikacijos ir tinklo konfigūracijos programa, pagrįsta FastAPI karkasu. Ši programa atlieka kelias pagrindines funkcijas: transporto priemonių autentifikavimą, jų skaitmeninių sertifikatų patikrą, WireGuard ryšio parametrų generavimą bei valdymą.

Vienas iš svarbiausių saugumo aspektų – tai transporto priemonių pateiktų skaitmeninių sertifikatų tikrinimas. RSU serveris leidžia registruotis tik toms transporto priemonėms, kurių sertifikatai yra patvirtinami naudojant lokaliai saugomą sertifikavimo centro (CA) šakninį sertifikatą. Registracijos metu transporto priemonė per TLS protokolu apsaugotą API galutinį tašką pateikia savo viešąjį raktą ir skaitmeninį sertifikatą.

Sėkmingai patikrinus sertifikatą, RSU serveris:

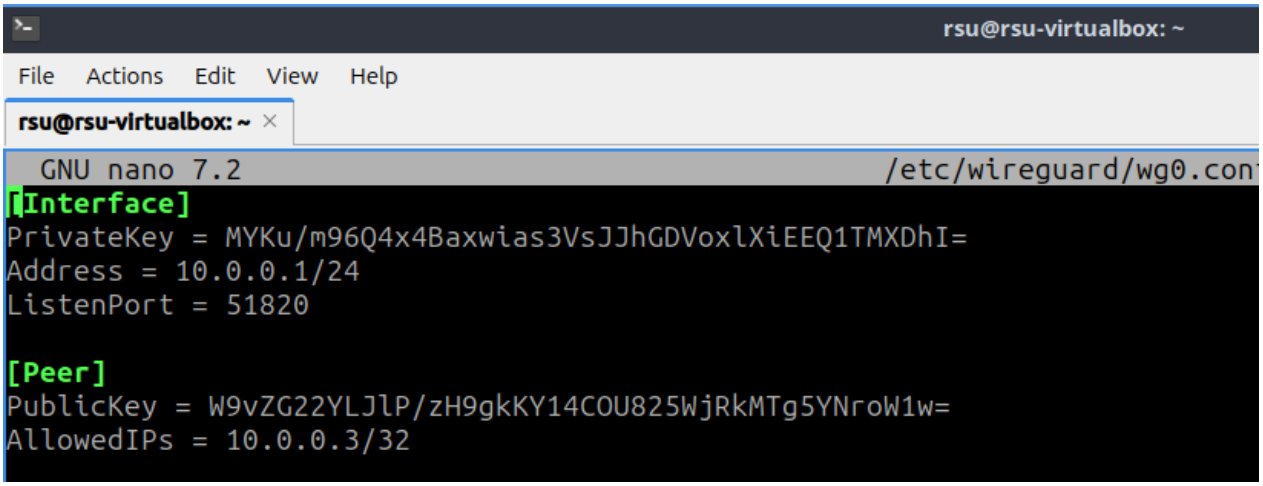
- Sugeneruoja transporto priemonei skirtą virtualaus privataus tinklo IP adresą.
- Dinamiškai prideda šį adresą ir pateiktą viešąjį raktą prie aktyvios WireGuard sąsajos naudodamas komandą *wg set*.
- Jei reikia, atnaujina ar paleidžia WireGuard sąsają naudodamas *wg-quick up wg0* arba analogiškus metodus.

WireGuard konfigūracija kiekvienai transporto priemonei sudaroma iš jos viešojo rakto, IP adreso (parametras *AllowedIPs*). Taip pat naudojamas parametras *PersistentKeepalive*, užtikrinantis pastovų ryšį su mobilia transporto priemone.

Visi šie parametrai yra taikomi tik aktyviai sąsajai, todėl jei reikia ilgalaikio išsaugojimo, serveris gali naudoti *wg-quick save wg0* komandą, kad automatiškai atnaujintų *wg0.conf* failą. Tačiau bazinėje prototipo versijoje ši konfigūracija lieka tik darbinėje atmintyje ir nėra įrašoma į disko konfigūracinius failus, todėl laikoma dinamine.

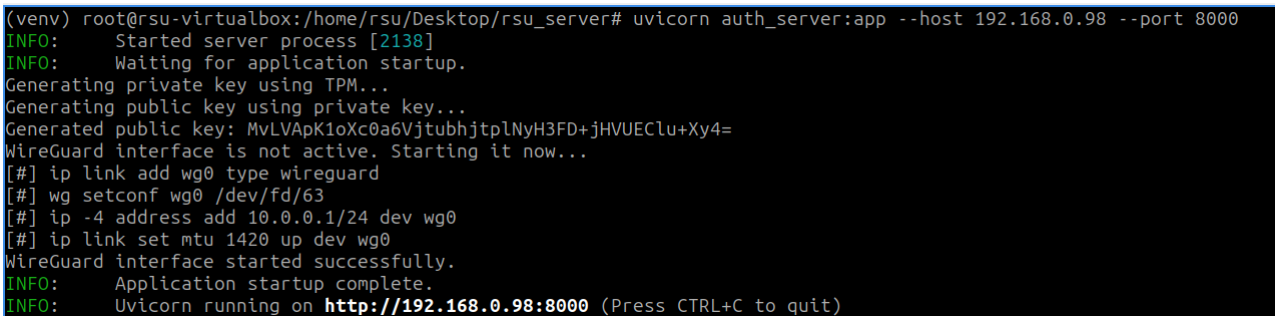
RSU autentifikacijos tarnyba geba vienu metu valdyti kelias transporto priemones, kiekvieną jų įtraukiant į WireGuard tunelį kaip atskirą tinklo mazgą. Dinaminis konfigūracijos atnaujinimas leidžia išvengti rankinio įsikišimo, todėl sistema tinkama naudoti realiomis sąlygomis.

Visa ši konfigūracija pateikiama ir praktiniame pavyzdyje. Žemiau parodytas sugeneruotas RSU serverio *wg0.conf* konfigūracijos failas, kuriame matomas RSU privatusis raktas, transporto priemonei priskirtas IP adresas bei jos viešasis raktas (žr. 18 pav.).



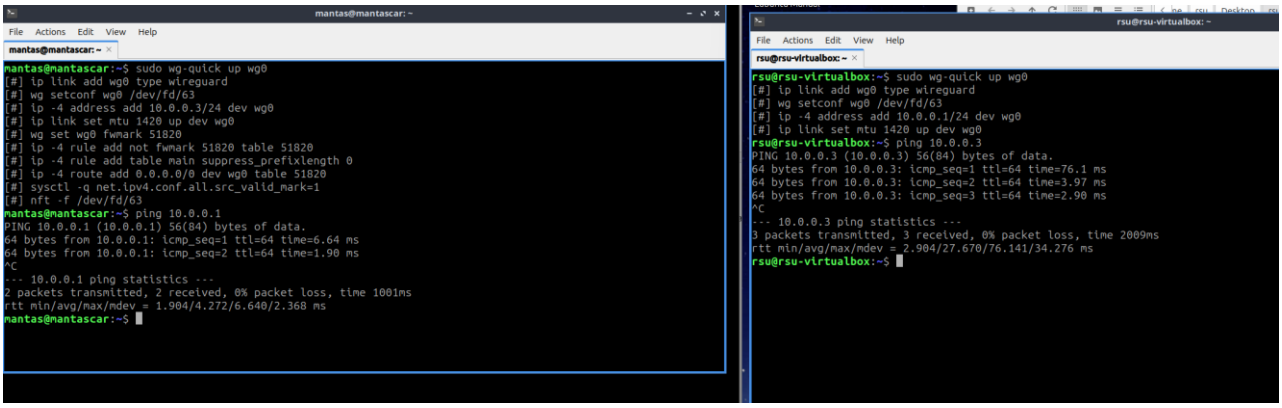
18 pav. Pavyzdinė WireGuard konfigūracijos ištrauka

Įjungiant kelio infrastruktūros objekto autentifikacijos serverį, įgalinama ir WireGuard virtualaus privataus tinklo sąsaja. Demonstracinis pavyzdys pateikiamas žemiau (žr. 19 pav.).



19 pav. WireGuard sąsajos įgalinimas serverio pusėje

Toliau pateikiamas demonstracinis ryšio užmezgimo pavyzdys, kuriame matoma, kaip abi pusės (RSU ir transporto priemonė) įgalina WireGuard sąsają, ją aktyvuoja ir atlieka ryšio patikrą tarpusavyje naudojant komandą *ping* (žr. 20 pav.). Iš terminalų išvesties akivaizdu, kad šifruotas virtualaus privataus tinklo ryšys sėkmingai veikia, o mazgai gali pasiekti vienas kitą vidiniame tinkle.



20 pav. Užmegztas WireGuard ryšys

Tolesniame poskyryje aptariama, kaip visa tai realizuojama transporto priemonės (CAR_1) pusėje nuo rakto generavimo iki šifruoto ryšio užmezgimo ir duomenų siuntimo.

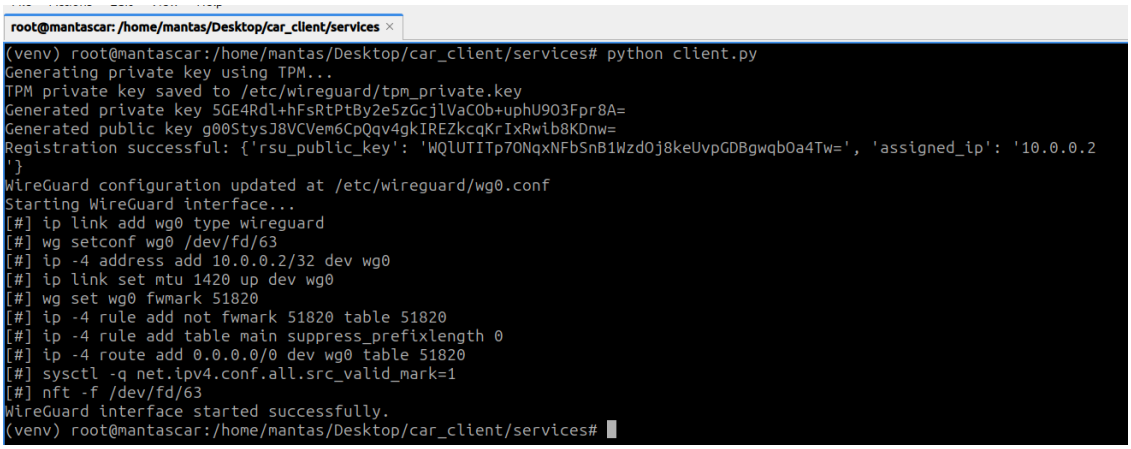
4.2. Autonominės transporto priemonės (CAR) veikimas

Autonominės transporto priemonės virtualioje mašinoje veikia Python kalba parašyta kliento autentifikacijos programa, kurios pagrindinis tikslas yra užmegzti saugų ryšį su kelio infrastruktūros objektu naudojant WireGuard tunelį ir TLS protokolu apsaugotą autentifikaciją. Šiame procese ypač svarbus TPM modulis, atsakingas už privačių raktų generavimą ir apsaugą.

Veikimo eiga susideda iš šių etapų:

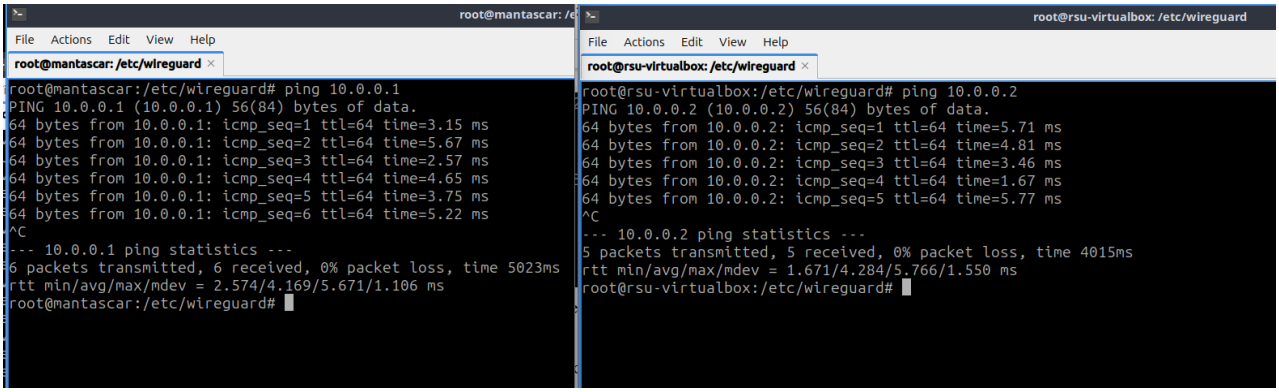
- **Privataus rakto generavimas naudojant TPM modulį**. Raktas sugeneruojamas ir užšifruojamas, o iš jo išvedamas viešasis raktas.
- **Autentifikacija per TLS protokolą**. Transporto priemonė siunčia RSU serveriui savo skaitmeninį sertifikatą ir viešąjį raktą.
- **WireGuard parametrų gavimas**. RSU serveris patikrina transporto priemonės atsiųstą skaitmeninį sertifikatą ir, jei jis galiojantis, siunčia atsakymą su autonominei mašinai priskirtu IP adresu bei savo WireGuard viešuoju raktu.
- **Konfigūracijos atnaujinimas ir tunelio aktyvavimas**. Kliento programa automatiškai suformuoja *wg0.conf* konfigūracijos failą, paleidžia WireGuard sąsają ir įgalina šifruotą virtualaus privataus tinklo tunelį.

Šis procesas demonstruojamas žemiau pateiktoje iliustracijoje, kurioje matoma terminalo išvestis nuo rakto generavimo iki sėkmingai paleisto šifruoto tunelio tarp transporto priemonės ir RSU (žr. 21 pav.).



21 pav. Autonominė transporto priemonė prisijungia prie RSU

Norint įsitikinti, kad VPT ryšys veikia tinkamai, atliekamas tinklo matomumo testas tarp autonominės transporto priemonės ir kelio infrastruktūros objekto. Šis testas patvirtina, kad abu mazgai pasiekiami vidiniame WireGuard tinkle (žr. 22 pav.).



22 pav. Tinklo pasiekiamumo patikrinimas

4.3. TPM modulio įtaka raktų saugumui

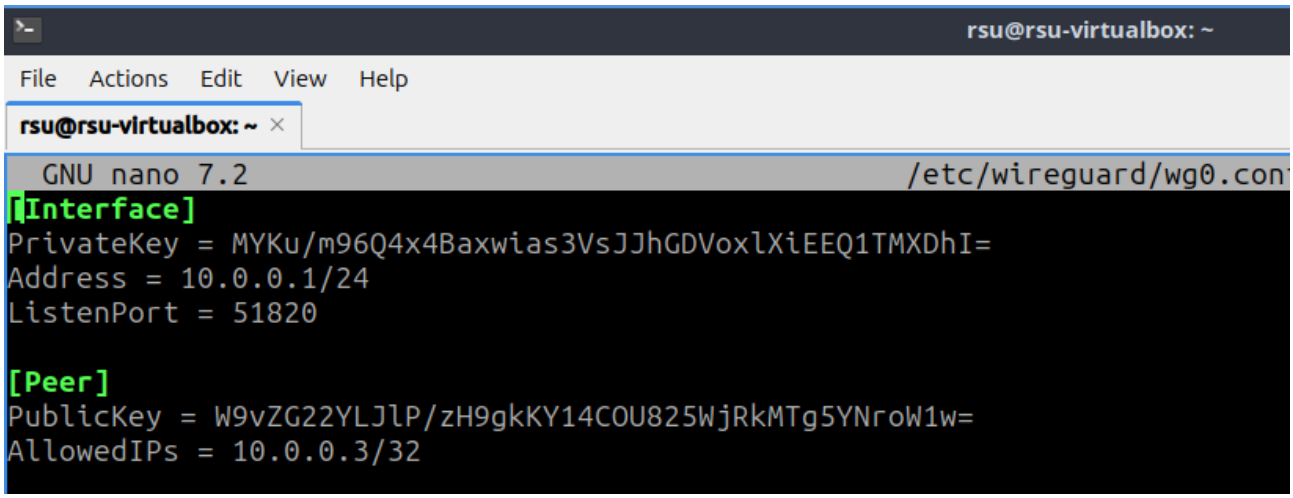
Šiame poskyryje atliekamas eksperimentinis palyginimas tarp dviejų prototipo versijų:

- **Be TPM**. WireGuard privatus raktas saugomas atviru tekstu sistemoje esančiame konfigūracijos faile.
- **Su TPM**. Raktas sugeneruojamas, užšifruojamas TPM ir iššifruojamas tik vykdymo metu.

4.3.1. Prototipas be TPM

Prototipe be TPM modulio integracijos, privatus raktas yra sugeneruojamas vieną kartą ir išsaugomas */etc/wireguard/wg0.conf* faile atviro teksto formatu (žr. 23 pav.).

Šis failas gali būti perskaitytas bet kurio naudotojo ar proceso, jei yra patenkama į autonominę transporto priemonę ir jei prieigos teisės leidžia skaityti šį failą. Raktą galima pasisavinti, o tai pažeistų sistemos vientisumą, nes tuomet galima išvesti WireGuard viešąjį raktą iš pasisavinto privataus rakto be papildomos autentifikacijos užklausos siuntimo RSU serveriui.



23 pav. Privataus rakto saugojimas atviro teksto formatu WireGuard konfigūracijos faile

4.3.2. Prototipas su TPM

Prototipo variante, kuriame naudojamas TPM modulis, WireGuard privataus rakto apsauga įgyvendinta vadovaujantis rekomenduojamomis kriptografinėmis praktikomis, kurios reikšmingai sustiprina bendrą sistemos saugumą. Skirtingai nei ankstesniame modelyje, kuriame privatus raktas buvo saugomas atviro teksto formatu konfigūracijos faile, šiame sprendime raktas neišsaugomas jokiuose nuolatiniuose diskuose ar viešai prieinamuose kataloguose. Raktas yra generuojamas vykdymo metu, naudojant TPM komandas, tokias kaip *tpm2_getrandom*, kurios leidžia gauti kriptografiškai saugų atsitiktinį baitų rinkinį. Tuomet ši reikšmė TPM modulyje yra užantspauduojama, sukuriant vadinamąjį užantspauduotą objektą (angl. *sealed object*), kuris saugomas .ctx failo formate. Šis failas negali būti perskaitytas ar iššifruotas be to paties TPM modulio, kuriame jis buvo sukurtas, o jo panaudojimas yra ribojamas specialiomis saugumo politikomis.

Vykstant autentifikacijos procesui, rakto išpakavimo (angl. *unseal*) operacija atliekama tik tiesiogiai TPM modulyje, o rakto reikšmė laikinai įkeliama tik į darbinę atmintį. Rakto vertė nėra įrašoma į failų sistemą atviro teksto formatu. Šis sprendimas užtikrina, kad net ir turint administratoriaus (*root)* lygio prieigą, privatusis raktas lieka apsaugotas, nebent būtų pažeistas pats TPM modulis. Tokiu būdu yra efektyviai pašalinama pagrindinė rizika, būdinga sistemoms be TPM – galimybė pasisavinti privatųjį raktą iš disko ir naudoti jį be jokios papildomos autentifikacijos ar autorizacijos.

TPM modulyje saugomi raktai pasižymi specifiniais saugumo atributais, kurie apibrėžia jų naudojimo kontekstą, galiojimo ribas bei veikimo apribojimus. Šie atributai nustatomi automatiškai rakto generavimo metu ir užtikrina, kad raktas būtų naudojamas tik saugioje bei kontroliuojamoje aplinkoje. Tyrimo metu sugeneruoti raktai turėjo šiuos atributus:

- **fixedtpm** – raktas galioja tik tame TPM įrenginyje, kuriame buvo sukurtas. Jo naudojimas kitame fiziniame TPM modulyje neįmanomas dėl fiksuoto ryšio su konkrečiu įrenginiu.
- **fixedparent** – raktas susiejamas su konkrečiu tėviniu raktu, todėl negali būti panaudotas kitose raktų hierarchijose ar su alternatyviais pagrindiniais raktais.

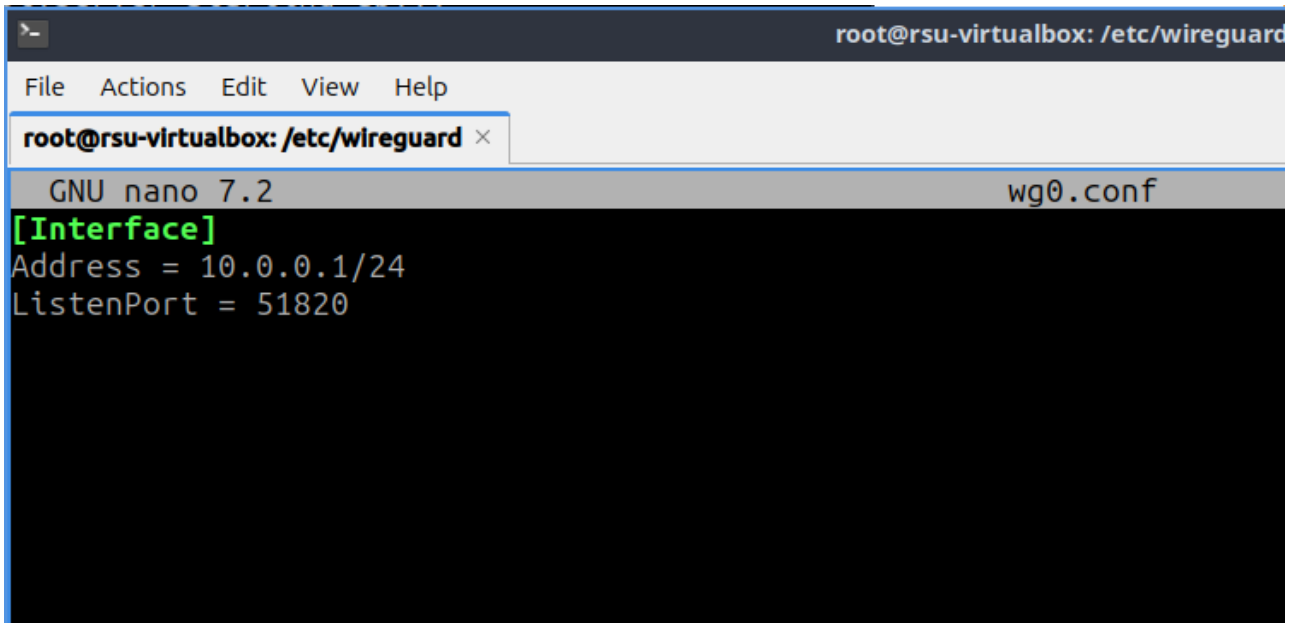
- **sensitivedataorigin** – garantuoja, kad privati rakto dalis buvo sugeneruota TPM viduje ir niekada nebuvo prieinama išorinei programinei įrangai ar naudotojui.
- **userwithauth** – nurodo, kad prieš naudojant raktą būtina autentifikacija, pavyzdžiui, pateikiant PIN kodą ar slaptažodį. Tai padeda užkirsti kelią neautorizuotam rakto naudojimui. Šiame prototipe autentifikacijos politika nebuvo nustatyta, todėl autentifikacija formaliai įvyksta TPM viduje. Vartotojui nereikia papildomai pateikti PIN kodo ar slaptažodžio. Pabrėžiama, kad šis mechanizmas suteikia galimybę ateityje įdiegti stipresnį autentifikavimą, jeigu to reikalautų saugumo politika.
- **restricted** – apibrėžia ribotą rakto paskirtį, leidžiant jį naudoti tik konkretiems veiksmams, pavyzdžiui, iššifravimui arba skaitmeniniam pasirašymui.
- **decrypt** – raktas gali būti naudojamas tik duomenų iššifravimo operacijoms. Jis nėra tinkamas šifravimui ar kitokioms kriptografinėms funkcijoms atlikti.

Žemiau pateikiamas užantspaudavimo pavyzdys, kuriame matomi išvardinti TPM modulio naudojami parametrai (žr. 24 pav.)

```
INFO:root:Server starting up...
INFO:root:Sealing WireGuard private key...
INFO:root:Creating new WireGuard private key...
name-alg:
  value: sha256
  raw: 0xb
attributes:
  value: fixedtpm|fixedparent|sensitivedataorigin|userwithauth|restricted|decrypt
  raw: 0x30072
type:
  value: rsa
  raw: 0x1
exponent: 65537
bits: 2048
```

24 pav. TPM užantspaudavimas

Taip pat akcentuotina tai, kad iššifruota WireGuard privataus rakto reikšmė į konfigūraciją įtraukiama dinamiškai vykdymo metu iš darbinės atminties, naudojant komandą *sudo wg set wg0 private-key /dev/stdin.* Tokiu būdu rakto reikšmė nėra įrašoma į failų sistemą. Ji yra perduodama tiesiogiai į aktyvią tinklo sąsają ir lieka tik darbinėje atmintyje. Dėl šios priežasties konfigūracijos faile *wg0.conf* nėra saugoma jokia tekstinė rakto informacija (žr. 25 pav.). Visgi, pažymėtina, kad paleidus sistemą iš naujo arba perkrovus tinklo sąsają, rakto reikšmė iš atminties pradingsta. Tokiu atveju, norint atkurti VPT tunelį, rakto iššifravimo ir pakartotinio įkėlimo procedūra turi būti vykdoma iš naujo, užtikrinant, kad autentifikacijos sistema turės prieigą prie TPM modulyje saugomo užantspauduoto objekto. Toks sprendimas reikšmingai padidina bendrą sistemos saugumą, nes apsaugo privatų raktą nuo ilgalaikio saugojimo bei galimų duomenų nutekėjimų.

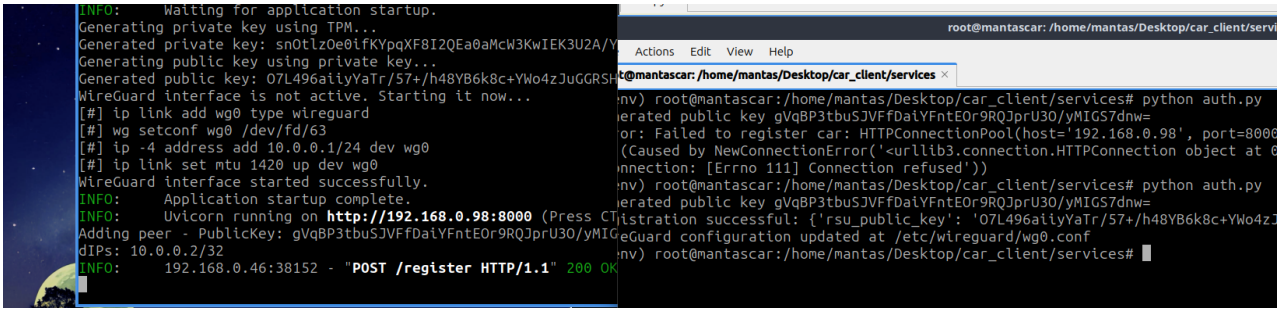


25 pav. WireGuard tinklo konfigūracinis failas

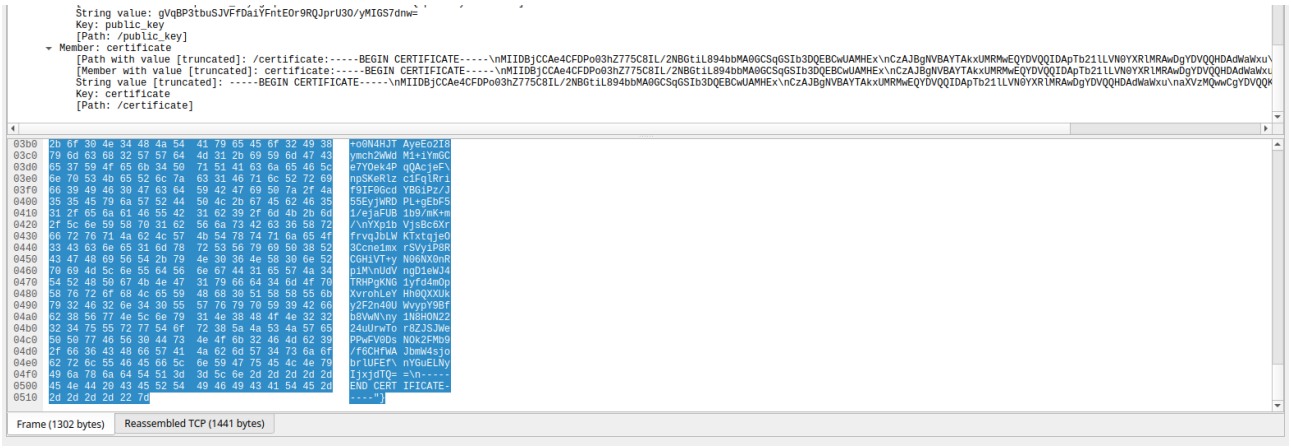
4.4. Saugus autonominės transporto priemonės autentifikacijos mechanizmas

Norint užtikrinti saugų autonominės transporto priemonės prijungimą prie kelio infrastruktūros objekto, būtina apsaugoti duomenų perdavimo procesą, ypač autentifikacijos etape. Vienas svarbiausių saugumo aspektų – tai transporto priemonės skaitmeninio sertifikato ir viešojo rakto perdavimas RSU serveriui. Šie duomenys privalo būti siunčiami naudojant TLS protokolą, kadangi neapsaugotas HTTP protokolas leidžia juos perimti bet kuriam tinklo stebėtojui, nes duomenys perduodami atviro teksto pavidalu.

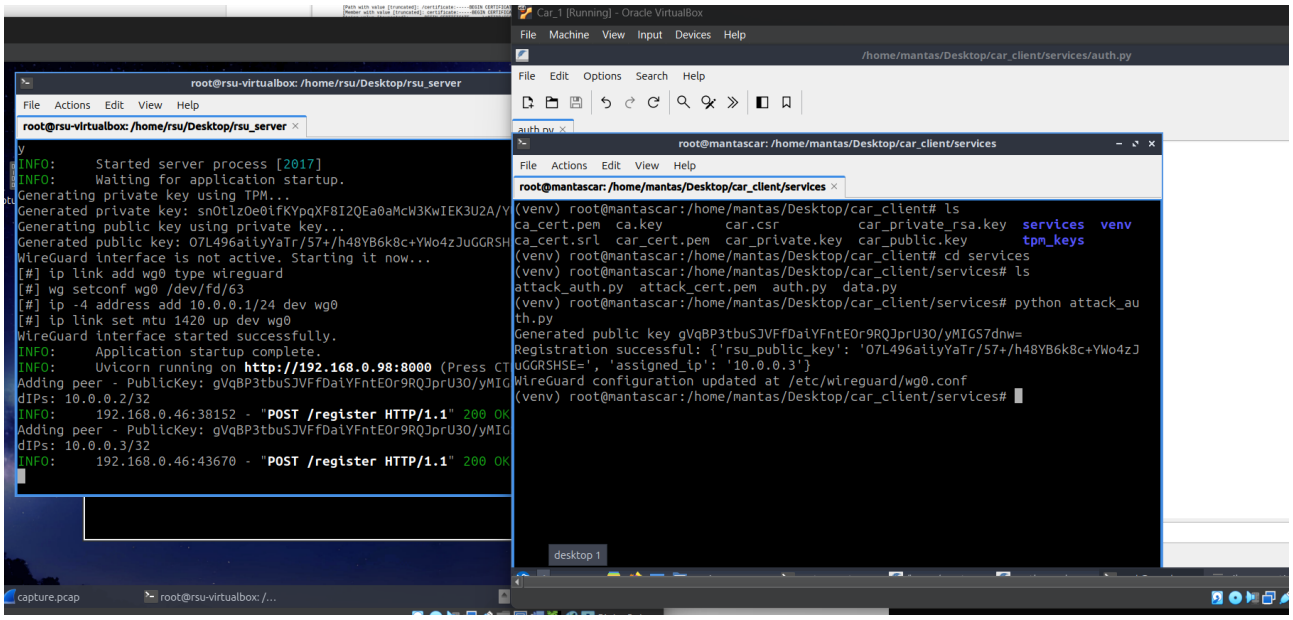
Demonstraciniame scenarijuje (žr. 26 pav.) transporto priemonė registruojasi prie RSU naudodama nesaugų HTTP protokolą. Ir nors registracijos metu terminale matoma, kad užklausa sėkmingai įvykdyta, analizė naudojant „Wireshark“ programinę įrangą parodė, kad visas sertifikato turinys perduodamas atviru JSON formatu (žr. 27 pav.). Toks elgesys yra pavojingas, nes kenkėjiška programa gali perimti šį sertifikatą, sugeneruoti savo raktų porą ir, apsimesdama teisėta transporto priemone, registruotis prie RSU. Ši grėsmė vizualizuota 28 paveiksle, kuriame pavaizduotas kenkėjiškas prisijungimas prie WireGuard virtualaus privataus tinklo.



26 pav. Transporto priemonė registruojasi nesaugiu HTTP protokolu



27 pav. Matomas tapatybės sertifikatas atviro teksto formatu

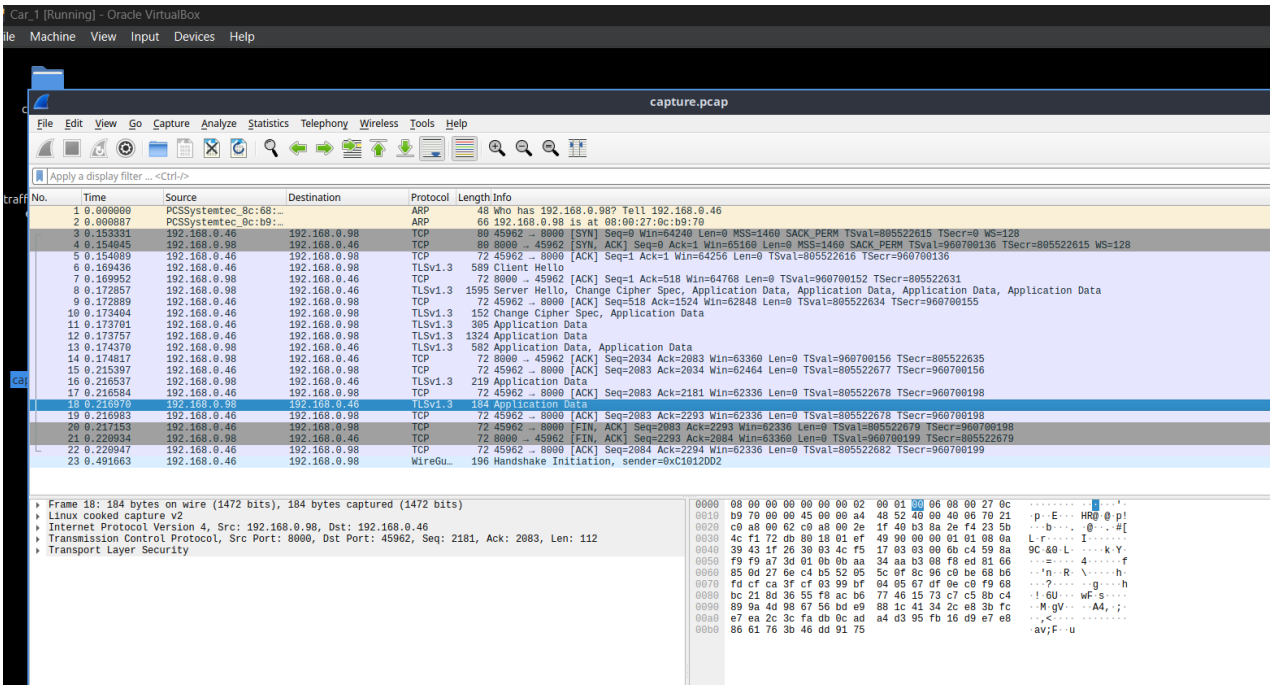


28 pav. Atakos scenarijus su pavogtu sertifikatu

Siekiant išvengti tokių saugumo pažeidimų, sukurtame prototipe autentifikacija tarp transporto priemonės ir RSU realizuojama naudojant TLS protokolą. TLS užtikrina duomenų konfidencialumą, vientisumą ir autentiškumą, neleisdamas trečiosioms šalims perimti ar pakeisti siunčiamų duomenų. Be to, protokolas palaiko atsitiktinių sesijų raktų generavimą, todėl net ir srauto perėmimo atveju neįmanoma atkurti ar pakartoti autentifikacijos. Šis mechanizmas užtikrina, kad tik teisėtos, skaitmeniniu sertifikatu patvirtintos transporto priemonės galėtų saugiai registruotis prie RSU, o jų ryšys su tinklu būtų apsaugotas nuo pasiklausymo ar klastojimo.

Pavyzdžiui, 29 paveiksle pateiktas tinklo srauto fragmentas rodo, kad TLS 1.3 protokolu apsaugoti duomenys matomi tik kaip dvejetainiai, užšifruoti duomenys. Taip apsaugomas skaitmeninis transporto priemonės sertifikatas bei viešasis raktas. Naudojant TLS, kiekviena sesija įgalinama su unikaliu sesijos raktu, todėl net ir perėmus šifruotą srautą, jo pakartotinis naudojimas ar iššifravimas

tampa sudėtingas. Šis metodas užtikrina duomenų konfidencialumą, vientisumą ir autentiškumą, o tai yra būtinos sąlygos saugiam autonominės transporto priemonės prijungimui prie tinklo.



29 pav. Šifruota autentifikacijos užklausa

4.5. Interneto pralaidumo matavimai

4.5.1. Interneto pralaidumo palyginimas tarp virtualaus privataus tinklo protokolų

Šiame poskyryje pateikiami eksperimentiniai duomenys, susiję su interneto srauto pralaidumo matavimu taikant skirtingus virtualaus privataus tinklo protokolus. Siekiama nustatyti, kaip kiekvienas virtualaus privataus tinklo protokolas paveikia tinklo pralaidumą.

Testai buvo atlikti naudojant dvi virtualiąsias mašinas tame pačiame fiziniame tinkle. Viena virtuali mašina veikė kaip iPerf3 serveris, kita kaip klientas. Abi mašinos turėjo identiškus techninius resursus: 4 procesoriaus branduoliai, 8 GB darbinės atminties ir Lubuntu operacinė sistema. Bandymai atlikti LAN aplinkoje siekiant sumažinti išorinių veiksnių įtaką rezultatams.

Naudoti virtualaus privataus tinklo protokolai ir konfigūracijos:

- **OpenVPN**. Paleistas naudojant standartinę UDP konfigūraciją (prievadas 1194), su AES-256-CBC šifravimu ir TLS pagrįsta autentifikacija. Naudoti šie konfigūracijos parametrai (žr. 1 ir 2 priedus):
  - `remote-cert-tls server` – reikalauja, kad serveris pateiktų galiojantį TLS sertifikatą, taip užtikrinant autentifikaciją.
  - `cipher AES-256-CBC` – nurodo simetrinio šifravimo algoritmą, kuriam naudojamas 256 bitų raktas ir CBC (angl. *Cipher Block Chaining*) režimas.

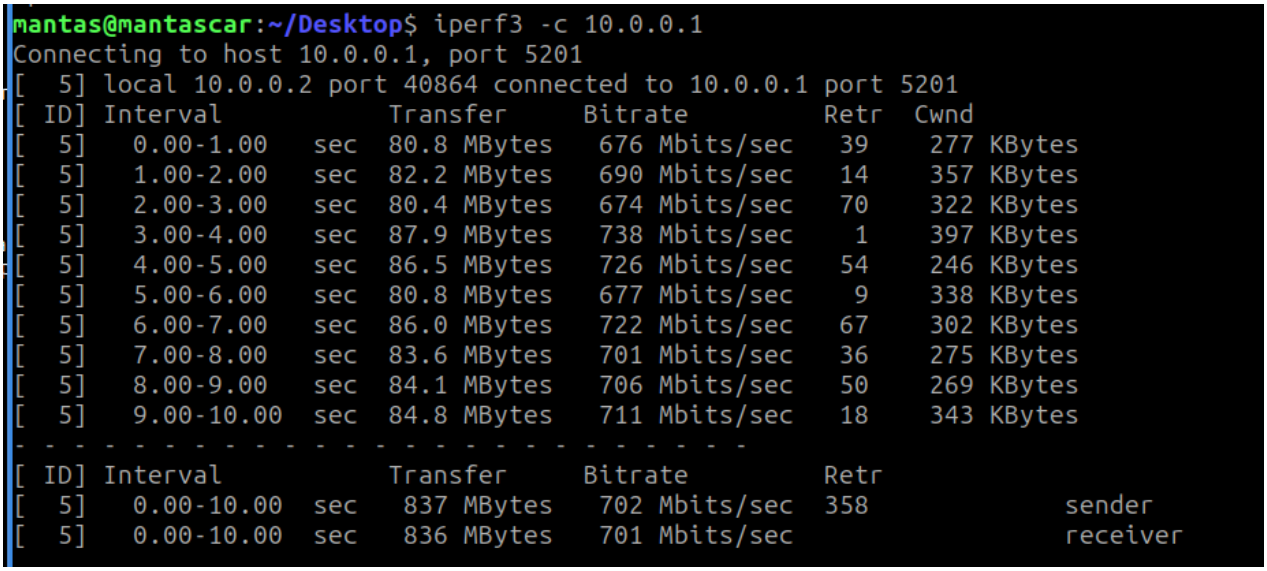
- `data-ciphers AES-256-GCM:AES-128-GCM` – nurodo pažangesnius šifrus naudojant GCM (*Galois/Counter Mode*), kuris vienu metu užtikrina tiek šifravimą, tiek duomenų autentiškumo tikrinimą.
- `data-ciphers-fallback AES-256-CBC` – nurodo atsarginį šifravimo metodą, kai GCM nėra palaikomas.
- `auth SHA256` – nustato HMAC (angl. *Hash-based Message Authentication Code*) su SHA-256 maišos funkcija, skirtą duomenų vientisumui ir autentifikacijai.
- `persist-key` – leidžia išlaikyti šifravimo raktus net ir atnaujinus ryšį.
- `persist-tun` – palaiko tunelio sąsają po nutraukimo ar pakartotinio prisijungimo.
- `verb 3` – nustato žurnalo (angl. *log*) išsamumo lygį. Reikšmė 3 reiškia vidutinį išsamumą.

- **IKEv2/IPsec.** Naudotas IKEv2 (angl. *Internet Key Exchange version 2*) protokolas su slaptažodžiu pagrįsta autentifikacija ir AES-256 šifru. Tunelis automatiškai įgalinamas bei palaikomas naudojant DPD (angl. *Dead Peer Detection*) mechanizmą, kuris leidžia periodiškai tikrinti, ar kita tunelio pusė vis dar pasiekiama. Nutrūkus ryšiui, tunelis automatiškai nutraukiamas, o sistemos ištekliai – atlaisvinami. Naudoti šie konfigūracijos parametrai:
  - `authby=psk` – nurodo, kad autentifikacijai naudojamas iš anksto bendrinamas slaptažodis.
  - `keyexchange=ikev2` – nurodo, kad naudojamas IKEv2 protokolas, skirtas saugiam raktų mainų ir VPT tunelio įgalinimo procesui tarp dviejų šalių.
  - `left=192.168.0.46` – nurodo kliento IP adresą.
  - `right=192.168.0.98` – nurodo serverio IP adresą.
  - `ike=aes256-sha256-modp1024!` – nurodo raktų mainams naudojamus algoritmus: AES-256, SHA-256 ir MODP1024.
  - `esp=aes256-sha256!` – nurodo duomenų šifravimo algoritmus: AES-256 ir SHA-256.
  - `rightsubnet=0.0.0.0/0` – nurodo, kad visas srautas bus maršrutizuojamas per VPT tunelį.
  - `auto=start` – nurodo, kad VPT tunelis bus įgalintas automatiškai paleidžiant IPsec paslaugą.
  - `dpdaction=clear` – nurodo, kad nutrūkus ryšiui, tunelis bus uždaromas, o ištekliai atlaisvinami.
  - `dpddelay=30s` – nurodo, kad DPD užklausos siunčiamos kas 30 sekundžių.
- **WireGuard.** Tunelio šifravimui naudojami numatytieji WireGuard protokolo algoritmai: ChaCha20 simetrinis šifravimo algoritmas, bei Poly1305 autentifikavimo algoritmas, užtikrinantis perduodamų duomenų vientisumą ir autentiškumą

Norint įvertinti skirtingų VPT protokolų įtaką tinklo pralaidumui, buvo pasitelktas iPerf3[2] – atvirojo kodo įrankis, skirtas tinklo greičio matavimui tarp dviejų tinklo mazgų. Šis įrankis palaiko tiek TCP,

[2] https://iperf.fr/

tiek UDP srauto generavimą, todėl leidžia atlikti patikimus bei tikslius pralaidumo testus nepriklausomai nuo naudojamo VPT sprendimo. Žemiau pateikiama iPerf3 terminalo išvestis, gauta matuojant duomenų perdavimo spartą per WireGuard tunelį tarp dviejų tinklo mazgų (žr. 30 pav.).

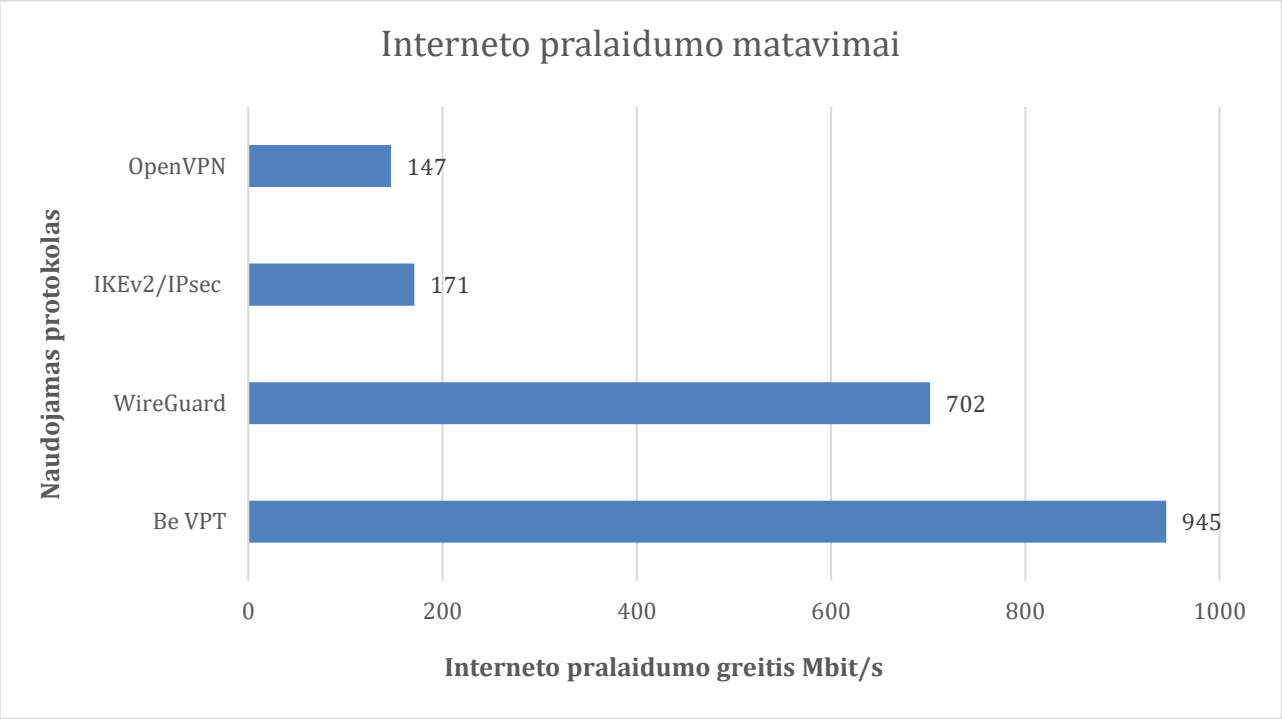


30 pav. Tinklo pralaidumo matavimas naudojant WireGuard

Matavimo metu vienas iš tinklo mazgų buvo naudojamas kaip serveris, o antrasis kaip klientas, siunčiantis duomenų srautą TCP protokolu. Tokiu būdu buvo fiksuojamas duomenų perdavimo greitis, retransliacijų skaičius bei bendras ryšio stabilumas. Rezultatai patiekiami 1 lentelėje bei pavaizduoti grafiškai (žr. 31 pav.).

1 lentelė. Interneto pralaidumo matavimų duomenys

Naudojamas protokolas	**Interneto pralaidumo greitis Mbit/s**	**Retransliacijų skaičius**
Be VPT	945	819
WireGuard	702	358
IKEv2/IPsec	171	1550
OpenVPN	147	224



31 pav. Interneto pralaidumo matavimai

Remiantis pateiktais eksperimentiniais rezultatais, galima daryti kelias reikšmingas išvadas apie VPT protokolų įtaką tinklo pralaidumui bei ryšio kokybei.

Didžiausias pralaidumo greitis buvo pasiektas be virtualaus privataus tinklo tunelio 945 Mbit/s, tačiau pastebėtinas ir netikėtai aukštas retransliacijų skaičius, siekiantis 819 Mbit/s. Tokia vertė gali būti susijusi su virtualizacijos aplinkos ypatumais, tinklo prievadų suderinamumu, aparatinės įrangos paravirtualizacija arba per dideliu didžiausiu tinklo paketo perdavimo vienetu (angl. *Maximum transmision size)* MTU dydžiu, dėl kurio kai kurie paketai gali būti dalinami į kelis arba atmetami. Šis reiškinys ypač tikėtinas naudojant TCP protokolą, kuris natūraliai atlieka retransliacijas siekdamas patikimo duomenų perdavimo.

WireGuard protokolas pasiekė 702 Mbit/s pralaidumą. Tai yra apie 26 % sumažėjimas, lyginant su ryšiu be virtualaus privataus tinklo. Vis dėlto, retransliacijų skaičius sumažėjo daugiau nei perpus ir siekė 358 Mbit/s. Tai rodo stabilų ir efektyvų duomenų tuneliavimą, nepaisant šifravimo apkrovos. Tokie rezultatai patvirtina WireGuard protokolo efektyvumą tiek pralaidumo, tiek patikimumo atžvilgiu.

IKEv2/IPsec ir OpenVPN protokolai parodė ženkliai mažesnį pralaidumą: 171 Mbit/s ir 147 Mbit/s. Sumažėjimas siekė kiek daugiau nei 80 %, lyginant su ryšiu be virtualaus privataus tinklo. Be to, IKEv2/IPsec retransliacijų skaičius pasiekė net 1550 Mbit/s, o tai gali rodyti prastą protokolo optimizaciją konkrečioje virtualioje testavimo aplinkoje arba didesnį jautrumą MTU neatitikimams ir tinklo nestabilumui. OpenVPN retransliacijų skaičius buvo mažesnis ir siekė 224 Mbit/s, bet pralaidumo požiūriu, OpenVPN buvo lėčiausias iš visų nagrinėtų protokolų.

4.6. Apsaugos metodų našumo palyginimas

Šiame poskyryje pateikiama eksperimentinė analizė, kuria siekta įvertinti skirtingų duomenų apsaugos metodų įtaką interneto pralaidumui, imituojant autonominės transporto priemonės ir kelio infrastruktūros objekto tarpusavio komunikaciją. Eksperimento metu buvo lyginamas siūlomas saugaus ryšio metodas, pagrįstas WireGuard virtualaus privataus tinklo tuneliu, su kriptografinėmis apsaugos priemonėmis, dažniausiai taikomomis V2X komunikacijos kontekste.

Pasirinkti kriptografiniai metodai apėmė:

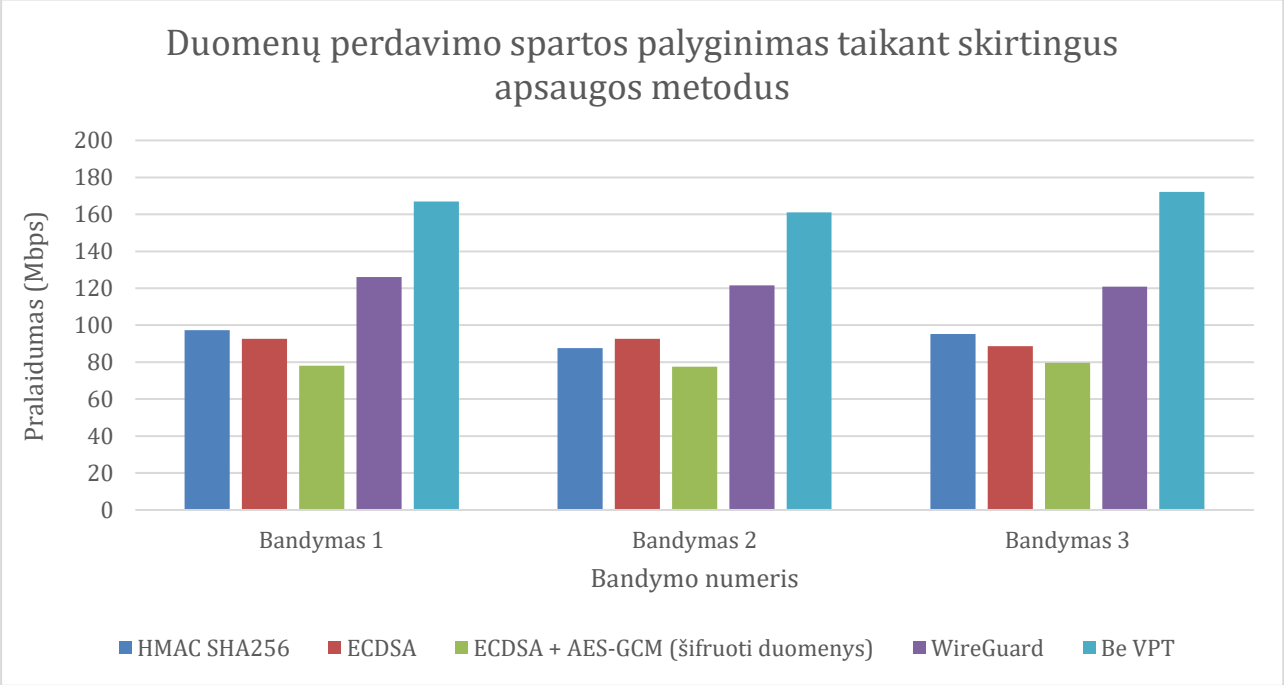
- atviro teksto perdavimą (be papildomos apsaugos);
- HMAC-SHA256 žinučių autentifikavimą;
- ECDSA skaitmeninį pasirašymą;
- ECDSA kartu su AES-GCM duomenų šifravimu;
- WireGuard tunelio naudojimą.

Tyrimo metu kiekviena iš šių apsaugos priemonių buvo taikoma perduodant 10 MB dydžio duomenų paketą, naudojant Python kalba parašytą kliento ir serverio aplikaciją, veikiančią per TCP lizdą (*socket*). Kiekvienam režimui buvo nustatyta vidutinė duomenų perdavimo sparta bei, kai taikyta, kriptografinio parašo generavimo trukmė. Gauti rezultatai pateikti 2 lentelėje.

2 lentelė. Skirtingų duomenų apsaugos metodų našumo palyginimas

	Bandymas 1 (Mbps)	Bandymas 2 (Mbps)	Bandymas 3 (Mbps)	Vid. paketo pasirašymo laikas (s)
HMAC SHA256	97.39	87.61	95.24	0.0390
ECDSA	92.56	92.56	88.67	90.82
ECDSA + AES-GCM (šifruoti duomenys)	78.07	77.53	79.56	0.0465
WireGuard	126.00	121.48	120.84	–
Be VPT	166.96	161.06	172.14	–

Gauti rezultatai taip pat pavaizduoti grafike, pateiktame 32 paveiksle.



32 pav. Duomenų perdavimo spartos palyginimas taikant skirtingus apsaugos metodus

4.6.1. WireGuard ir V2X metodų našumo palyginimas

Eksperimentas parodė, kad kriptografiniai pasirašymo ir šifravimo procesai turi aiškų poveikį duomenų perdavimo spartai. Palyginus, HMAC SHA256 turėjo mažiausią našumo kritimą tarp kitų V2X komunikacijoje naudojamų metodų, o ECDSA kartu su AES-GCM šifravimu sumažino pralaidumą daugiau nei 25 % dėl papildomos skaičiavimo naštos.

Tuo tarpu WireGuard virtualaus privataus tinklo tunelis pasirodė efektyvesnis nei bet kuris parašu pagrįstas metodas, o tiesioginis perdavimas be VPT užtikrino didžiausią pralaidumą. Šis metodas nenaudojo papildomo paketų pasirašymų ar duomenų šifravimo naudojant įprastus V2X algoritmus, nes patys tinklo paketai jau šifruojami ChaCha20-POLY1305 algoritmu.

Šie rezultatai parodo, kad kompromisai tarp saugumo ir našumo yra neišvengiami, ypač taikant duomenų pasirašymo ar šifravimo mechanizmus realaus laiko scenarijuose. Vis dėlto, WireGuard protokolas išsiskiria efektyviu šifravimo algoritmų įgyvendinimu ir pasižymi itin geru balansu tarp saugumo, našumo bei tinklo pralaidumo, todėl jis gali būti vertinamas kaip perspektyvus sprendimas, leidžiantis efektyviau išnaudoti šiuolaikinius kriptografinius algoritmus, nepakenkiant tinklo paketų perdavimo spartai.

Pastebėtina, kad šio eksperimento metu pasiekti pralaidumo rodikliai (iki ~166 Mbit/s be VPT ir ~122 Mbit/s naudojant WireGuard protokolą) yra mažesni nei anksčiau fiksuoti maksimalūs rezultatai naudojant iPerf3 įrankį (virš 700 Mbit/s). Toks skirtumas gali būti paaiškinamas tuo, jog šiame teste duomenų perdavimas vyko per Python pagrindu sukurtą programą, kurioje papildomos operacijos, tokios kaip JSON duomenų paruošimas, parašo generavimas ar šifravimas, turėjo įtakos bendrai sisteminei apkrovai. Be to, perdavimui naudotas TCP lizdas ir realus aplikacinio sluoksnio

apdorojimas, o ne žemas tinklo sluoksnis, kaip yra iPerf3 atveju. Nepaisant to, šio eksperimento metu buvo nuosekliai stebimi procentiniai našumo skirtumai tarp skirtingų apsaugos metodų, todėl rezultatai išlieka tinkami tarpusavio palyginimui ir leidžia vertinti kiekvieno metodo poveikį realiomis komunikacijos sąlygomis.

Apibendrinant galima teigti, kad WireGuard protokolas pasižymi puikia pusiausvyra tarp saugumo ir našumo. Tuo tarpu V2X komunikacijos metodai, kuriuose taikomas kriptografinis pasirašymas ir (arba) duomenų šifravimas, sukelia didesnę skaičiavimo apkrovą. Todėl realaus laiko scenarijuose tokių sprendimų taikymui būtinas atidus architektūrinis planavimas, siekiant išlaikyti priimtiną tinklo vėlavimą ir duomenų perdavimo spartą.

4.7. Tyrimo išvados

Apibendrinant eksperimentinę dalį, galima teigti, kad pasiūlytas saugios komunikacijos sprendimas, pagrįstas WireGuard protokolu, TPM pagrindu veikiančiu raktų valdymu bei TLS autentifikacija, užtikrina efektyvų balansą tarp saugumo, našumo bei sistemos valdymo funkcijų. Tyrimas atskleidė, kad klasikiniai virtualaus privataus tinklo sprendimai, tokie kaip OpenVPN ar IKEv2/IPsec, žymiai sumažina tinklo paketų pralaidumą, o kriptografiniai V2X pasirašymo metodai reikalauja papildomų sistemos resursų. Modernių technologijų, tokių kaip TPM, dinaminio raktų įkėlimo ir optimizuotų šifravimo algoritmų, taikymas leidžia išlaikyti aukštą saugumo lygį neprarandant sistemos našumo net realiomis veikimo sąlygomis. Gauti rezultatai pagrindžia sukurto sprendimo potencialą tolimesniam taikymui ir plėtrai autonominių transporto priemonių komunikacijos kontekste.

Išvados ir rezultatai

1. Atlikus autonominių transporto priemonių tarpusavio komunikacijos ir saugumo reikalavimų analizę, nustatyta, kad dėl didelio mobilumo ir nepastovios infrastruktūros būtina naudoti sprendimus, atitinkančius realaus laiko veikimo sąlygas, pasižyminčius maža skaičiavimo apkrova bei atsparumu trikdžiams.

2. Atlikta taikomų saugumo sprendimų analizė atskleidė, kad šiuolaikinių transporto sistemų saugiam duomenų perdavimui ir programinės įrangos integralumui užtikrinti efektyviausiai veikia kompleksinis pažangių kriptografinių sprendimų taikymas. ChaCha20-Poly1305 algoritmas, plačiai naudojamas TLS 1.3 ir WireGuard protokoluose, pasižymi optimaliu simetrinio šifravimo ir autentifikacijos derinimu, kas ypač svarbu realaus laiko komunikacijoje transporto sistemose. Patikimai V2X komunikacijai užtikrinti pasirinktas ECDSA skaitmeninių parašų mechanizmas, kuris ne tik efektyviai identifikuoja siuntėją, bet ir garantuoja perduodamų pranešimų autentiškumą. Saugiam raktų apsikeitimui ir skaitmeninių sertifikatų patikrai gali būti naudojamas TLS protokolas, kuris užtikrina saugų pradinio ryšio užmezgimą. Tyrimo metu nustatyta, kad privačių kriptografinių raktų apsaugai tikslinga integruoti TPM modulį. Taip privatūs raktai nebūtų saugomi atviru tekstiniu formatu, o jų naudojimas apribotas saugiomis TPM procedūromis ir užantspauduotais objektais. Šis optimizuotas saugumo metodų derinys atitinka aukščiausius transporto sistemų saugumo reikalavimus ir sudaro patikimą pagrindą tolimesniam transporto sistemų vystymui.

3. Eksperimentinis tyrimas, atliktas naudojant virtualių transporto priemonių ir infrastruktūros sąveikos modelį, parodė, kad siūlomas metodas leidžia patikimai įgalinti saugų ryšį ir pasižymi aukštu duomenų perdavimo pralaidumu. Lyginant su tradiciniais virtualaus privataus tinklo sprendimais, tokiais kaip OpenVPN ar IKEv2/IPsec, WireGuard protokolas pasižymėjo geresniu našumu. Tuo tarpu V2X parašų mechanizmai, nors ir užtikrina aukštą saugumo lygį, yra skaičiavimo požiūriu brangūs ir reikalauja didesnių resursų. Siūlomas metodas sėkmingai suderina saugumą, našumą ir rakto apsaugos patikimumą, todėl yra tinkamas naudoti realiose autonominių transporto sistemų aplinkose.

4. Atsižvelgiant į gautus rezultatus, rekomenduojama siūlomą metodą taikyti autonominių transporto sistemų scenarijuose, kuriuose būtina užtikrinti nepertraukiamą, saugų ir efektyvų ryšio užmezgimo procesą tarp judančių mazgų ir infrastruktūros komponentų. Dėl suderinto saugumo, našumo ir rakto apsaugos balanso šis metodas ypač tinkamas realaus laiko taikymams, pavyzdžiui, transporto priemonių registracijai prie RSU mazgų, kritinės informacijos perdavimui eismo valdymo situacijose ar komunikacijai autonominių transporto priemonių grupių formavimo metu. Ateityje metodas gali būti toliau plėtojamas integruojant pažangesnius autentifikacijos procesus, tinklo apkrovos balansavimo mechanizmus bei išmaniąją sertifikatų gyvavimo ciklo priežiūrą.

Literatūros sąrašas

[1] M. Placek, „Projected global sales of autonomous vehicles 2019–2030“, *Statista*. 2022. [interaktyvus], [žiūrėta 2023-01-13]. Prieiga per: https://www.statista.com/statistics/1230733/projected-sales-autonomous-vehicles-worldwide/

[2] Ani Petrosyan, „Most reported cybercrime in the U.S. 2023, by number of individuals affected“, *Statista*. 2025. [interaktyvus], [žiūrėta 2025-05-18]. Prieiga per: https://www.statista.com/statistics/184083/commonly-reported-types-of-cyber-crime-global/

[3] M. Armstrong, „The Internet Crime Business is Booming“, *Statista*. 2022. [interaktyvus], [žiūrėta 2023-01-13]. Prieiga per: https://www.statista.com/chart/20845/financial-losses-suffered-by-victims-of-internet-crimes/

[4] J. Monteagudo, „Cyber Security for Connected and Autonomous Vehicles“, *Cyber Startup Observatory*. [interaktyvus], [žiūrėta 2023-01-13]. Prieiga per: https://cyberstartupobservatory.com/cyber-security-connected-autonomous-vehicles/

[5] T. Yeferny, S. Hamad, „Vehicular Ad-hoc Networks: Architecture, Applications and Challenges“, *International Journal of Computer Science and Network Security,* vol. 20, nr. 2, p. 1–10, 2020. [interaktyvus], [žiūrėta 2023-01-13]. Prieiga per: https://www.researchgate.net/publication/340085318

[6] S. S. Shah, A. Malik, A. U. Rahman, „Time Barrier-Based Emergency Message Dissemination in Vehicular Ad-hoc Networks“, *ResearchGate*. 2019. [interaktyvus], [žiūrėta 2023-01-13]. Prieiga per: https://www.researchgate.net/figure/VANET-communication-architecture_fig3_330772224

[7] W. Ahsan, M. F. Khan, F. Aadil, M. Maqsood, S. Ashraf, Y. Nam, S. Rho, „Optimized Node Clustering in VANETs by Using Meta-Heuristic Algorithms“, *Electronics*. 2020. [interaktyvus], [žiūrėta 2023-01-14]. Prieiga per: https://www.mdpi.com/2079-9292/9/3/394

[8] A. Weimerskirch, „V2X Security & Privacy: The Current State and Its Future“, *escrypt Inc.* [interaktyvus], [žiūrėta 2025-05-09]. Prieiga per: https://www.weimerskirch.org/files/Weimerskirch_V2XSecurity.pdf

[9] X. Wang, „Research on ECDSA-Based Signature Algorithm in Blockchain“, *Finance and Market*. 2019. [interaktyvus], [žiūrėta 2025-05-09]. Prieiga per: https://www.researchgate.net/publication/342660617_Research_on_ECDSA-Based_Signature_Algorithm_in_Blockchain/fulltext/5eff23bb299bf18816fcf51a/Research-on-ECDSA-Based-Signature-Algorithm-in-Blockchain.pdf

[10] Y. Nir, A. Langley, „ChaCha20 and Poly1305 for IETF Protocols“, *RFC 7539*. 2015. [interaktyvus], [žiūrėta 2025-05-09]. Prieiga per: https://www.rfc-editor.org/rfc/rfc7539

[11] J. Thakkar, „TLS 1.3 Handshake: Taking a Closer Look“, *The SSL Store*. 2018. [interaktyvus], [žiūrėta 2025-05-18]. Prieiga per: https://www.thesslstore.com/blog/tls-1-3-handshake-tls-1-2/

[12] J. A. Donenfeld, „WireGuard: Next Generation Kernel Network Tunnel“, *wireguard.com*. 2020. [interaktyvus], [žiūrėta 2023-01-15]. Prieiga per: https://www.wireguard.com/papers/wireguard.pdf

[13] L. Osswald, M. Haeberle, M. Menth, „Performance Comparison of VPN Solutions“, *University of Tübingen*. [interaktyvus], [žiūrėta 2023-01-14]. Prieiga per: https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/100430/performance_comparison_of_vpn_solutions.pdf

[14] M. Zhu, J. Wang, „A Trusted Data Access and Transmission Method under Internal and External Threats in the Space Information Network“, *CFIMA '24: Proceedings of the 2024 2nd International Conference on Frontiers of Intelligent Manufacturing and Automation*. 2024. [interaktyvus], [žiūrėta 2025-01-14]. Prieiga per: https://dl.acm.org/doi/full/10.1145/3704558.3705531

[15] A. Ghosal, M. Conti, „Security Issues and Challenges in V2X: A Sruvę“, *Computer Networks*, vol. 169, 2020, art. nr. 107093. [interaktyvus], [žiūrėta 2025-02-12]. Prieiga per: https://doi.org/10.1016/j.comnet.2019.107093

[16] Z. Liu, „Application and Security Analysis of Virtual Private Network (VPN) in Network Communication“, *Academic Journal of Computing & Information Science*, vol. 6, nr. 11, p. 52–59, 2023. [interaktyvus], [žiūrėta 2025-02-15]. Prieiga per: https://www.francis-press.com/uploads/papers/VydclV52hOq5Qta6hQOQozgqGL1aIyHXdzzoDFvD.pdf

[17] H. da Rocha, P. Caruso, J. Pereira, P. Serra, A. E. Santo, „Discussion on Secure Standard Network of Sensors Powered by Microbial Fuel Cells“, *Sensors*, vol. 23, nr. 19, art. nr. 8227, 2023. [interaktyvus], [žiūrėta 2025-03-12]. Prieiga per: https://www.mdpi.com/1424-8220/23/19/8227

[18] G. Nookala, „The Role of SSL/TLS in Securing API Communications: Strategies for Effective Implementation“, *Journal of Computing and Information Technology*, vol. 4, nr. 1, 2024. [interaktyvus], [žiūrėta 2025-03-12]. Prieiga per: https://universe-publisher.com/index.php/jcit/article/view/20/20

[19] J. Zhou, W. Fu, W. Hu, Z. Sun, T. He ir Z. Zhang, „Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey“, *Electronics*, vol. 13, nr. 20, art. nr. 4000, 2024. [interaktyvus], [žiūrėta 2025-04-12]. Prieiga per: https://www.mdpi.com/2079-9292/13/20/4000

[20] C. Ryu, J.-H. Lee, D.-H. Kim, H.-S. Lee, Y.-S. Kim, J.-H. Han, J.-n. Kim, „A Comprehensive Survey of TPM for Defense Systems“, *KSII Transactions on Internet and Information Systems*, vol. 18, nr. 7, p. 1953–1967, 2024. [interaktyvus], [žiūrėta 2025-05-10]. Prieiga per: https://koreascience.kr/article/JAKO202425557628503.page

[21] F. Voron, „Building Data Science Applications with FastAPI: Develop, manage, and deploy efficient machine learning applications with Python“. *Packt Publishing*, 2023. [interaktyvus], [žiūrėta 2025-05-15]. Prieiga per: https://books.google.lt/books?id=NCXMEAAAQBAJ

Priedai

1 priedas. OpenVPN serverio konfigūracijos parametrai

```
port 1194

proto udp

dev tun

ca ca.crt

cert server.crt

key server.key

dh dh.pem

auth SHA256

tls-auth ta.key 0

topology subnet

server 10.8.0.0 255.255.255.0
```

2 priedas. OpenVPN kliento konfigūracijos parametrai

```
client

dev tun

proto udp

remote 192.168.0.98 1194

resolv-retry infinite

nobind

persist-key

persist-tun

remote-cert-tls server

auth SHA256

cipher AES-256-CBC

data-ciphers AES-256-GCM:AES-128-GCM

data-ciphers-fallback AES-256-CBC

key-direction 1

verb 3
```

3 priedas. IKEv2/IPsec konfigūracijos parametrai

```
config setup
    charondebug="ike 2, knl 2, cfg 2, net 2, esp 2, dmn 2, mgr 2"


conn ikev2-psk
    auto=start
    keyexchange=ikev2
    type=tunnel
    authby=psk
    left=192.168.0.46
    leftid=192.168.0.46
    leftsourceip=%config
    right=192.168.0.98
    rightid=192.168.0.98
    rightsubnet=0.0.0.0/0
    ike=aes256-sha256-modp1024!
    esp=aes256-sha256!
    dpdaction=clear
    dpddelay=30s
    dpdtimeout=120s
```